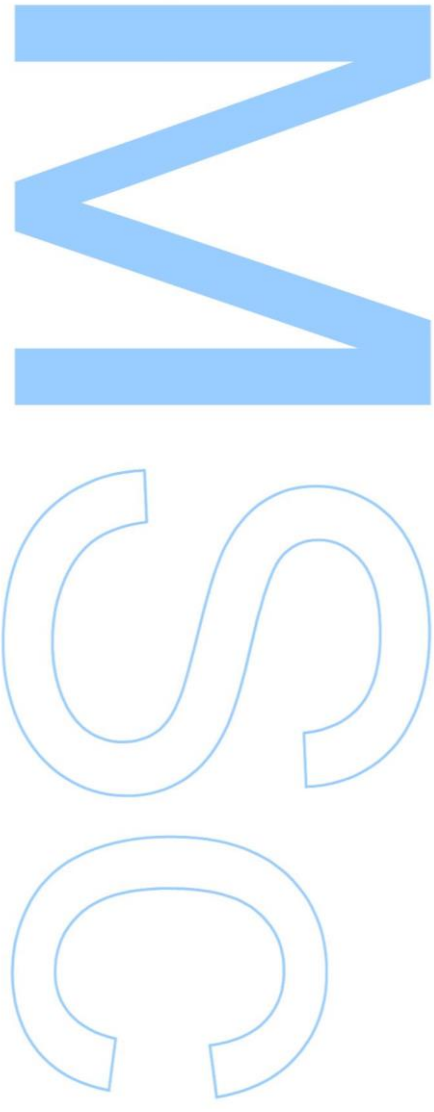
Análise da utilização dos recursos do Moodle para prever classificações

Bruno Miguel Ribeiro Cabral
Mestrado Integrado em Engenharia de Redes e Sistemas Informáticos
Departamento de Ciência de Computadores
2019

Orientador
Álvaro Pedro de Barros Borges Reis Figueira, Professor Auxiliar,
Faculdade de Ciências da Universidade do Porto



U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, _____/_____/_____

Resumo

Atualmente, os estudantes têm vindo gradualmente a ser avaliados usando uma plataforma *online*. Como tal os educadores têm tido a preocupação de alterar o seu método de ensino fazendo a transição do papel para o digital. O uso de um conjunto diversificado de perguntas, que variam desde questionários a questões abertas, é comum na maioria dos cursos universitários. Em muitos cursos, hoje, a metodologia de avaliação também promove a participação *online* dos estudantes em fóruns, o *download* e *upload* de arquivos modificados ou até mesmo a participação em atividades em grupo. Ao mesmo tempo, novas teorias pedagógicas que promovem a participação ativa dos estudantes no processo de aprendizagem, e o uso sistemático da aprendizagem baseada em problemas, estão a ser adotadas, usando um sistema de *e-learning* para essa finalidade. No entanto, embora possa haver um grande *feedback* dessas atividades para os estudantes, geralmente essa informação é restrita, e pouco útil se surgir muito tarde no curso.

Dai surge a necessidade de criar um modelo, capaz de prever qual será a classificação final de um estudante ao longo do semestre. Esta abordagem baseia-se no facto de que, ao obter essas informações no início do semestre, os estudantes e educadores podem ainda ter a oportunidade de resolver eventuais problemas relacionados com as ações *online* atuais do estudante em relação ao curso. No final é proposta a criação de uma metodologia que utiliza os registos recolhidos pelo Moodle, sobre as atividades dos estudantes de uma unidade curricular ao longo de três anos, para prever qual será a sua classificação final.

Foi feita uma análise dos dados inicias, concluindo-se que estes seriam insuficientes. Surgiu, assim, a necessidade de serem criados campos adicionais, contendo informação relativa à duração da sessão *online* e codificação das interações. Uma vez obtida toda a informação, foi realizada a limpeza e organização dos dados. Na fase seguinte, foram identificadas e criadas as variáveis independentes e uma variável objetivo. Estas variáveis foram usadas em conjunto com um algoritmo de Machine Learning para permitir a realização de previsões. Foram definidos os conjuntos de treino e de teste. O conjunto de treino foi usado como um método de aprendizagem supervisionado, para em conjunto com as variáveis, permitir criar uma árvore de decisão para realizar previsões. Já o conjunto de teste foi usado para obter os resultados e determinar qual a qualidade preditiva do modelo.

Os resultados obtidos pelo estudo, provam que este modelo é eficaz a realizar previsões de boa qualidade logo no início do semestre, sendo que à medida que o semestre decorre a qualidade preditiva do modelo aumenta. As variáveis independentes têm capacidade de generalização suficiente para poderem ser usadas em outros cursos *online*.

Abstract

Nowadays, students are gradually being assessed using an online platform. As such educators have been concerned to change their teaching methodology by making the transition from paper to digital. The use of a diverse set of questions, ranging from quizzes to open-ended questions, is common in most college courses. In many courses today, the assessment methodology also promotes student online participation in forums, downloading and uploading modified files, or even participating in group activities. At the same time, new pedagogical theories that promote students' active participation in the learning process and the systematic use of problem-based learning are being adopted using an e-learning system for that purpose. However, while there may be great feedback from these activities to students, this information is often restricted, and not useful if it comes too late in the course.

The need arose to create a model that can predict what a student's final grade will be during the semester. This approach is based on the fact that by obtaining this information at the beginning of the semester, students and educators may still have the opportunity to address any issues related to the student's current online actions regarding the course. In the end it is proposed, the creation of a methodology that uses the records collected by Moodle, about the activities of students of a course over a span of three years, to predict what will be their final grade.

An analysis of the intial data was performed, concluding that the information contained was insufficient. This conclusion led to creation of additional fields, including session length information regarding the duration of the online session and coding of the interactions. Once all the information was obtained, the data was cleaned and organized. In the next phase, the independent variables and an objective variable were identified and created. These variables were used in conjunction with a Machine Learning algorithm to allow predictions to be made. Training and test sets were defined. The training set was used as a supervised learning method, in conjunction with the variables, to create a decision tree to make predictions. The test set was then used to obtain the results and determine the predictive quality of the model.

The results of the study prove that the created model is effective in performing good quality predictions early in the semester, and as the semester elapses the predictive quality of the model increases. Independent variables have sufficient generalizability to be used in other online courses.

Agradecimentos

Este documento representa a conclusão do meu percurso académico. Como tal, quero agradecer a algumas pessoas que me influenciaram positivamente ao longo destes anos.

Aos meus pais Paulo e Isabel Cabral, por acreditarem em mim, por me amarem e apoiarem em todo o meu percurso académico ao longo dos anos, certamente sem eles não seria a pessoa que sou hoje.

Às minhas avós Maria José e Maria Alice e ao meu padrinho Armando Cabral, por me desejarem sempre o melhor e acreditarem nas minhas capacidades.

Ao meu avô, Ramiro Augusto Cabral, eterna saudade.

À minha namorada, Tânia Carvalho, obrigado por estares presentes nos bons e maus momentos, por me ajudares nas minhas dificuldades e no meu trabalho, acima de tudo obrigado por todo o carinho e apoio que me dás.

Ao meu orientador, o Prof. Álvaro Figueira, cujo vasto conhecimento, visão crítica e disponibilidade para me ajudar foram sempre um incentivo para eu fazer o meu melhor.

A gem cannot be polished without friction,
nor a man perfected without trials.

Lucius Annaeus Seneca
Marcus Aurelius

Dedico aos meus pais, avós, padrinho, namorada e avô.

Índice de Conteúdos

Lista de Tabelas

Lista de Figuras

Acrónimos

CRISP-DM *Cross-Industry Standard Process for Data Mining*

DCC Departamento de Ciência de Computadores

EDM *Educational Data Mining*

FCUP Faculdade de Ciências da Universidade do Porto

FN Falsos Negativos

FP Falsos Positivos

FILO *First In Last Out*

GI Gabinete de Informática

IA Inteligência Artificial

IGA *Intelligent Grading Agent*

IR *Information Retrieval*

LA *Learning Analytics*

LMS *Learning Management System*

NLP *Natural Language Processing*

SVM *Support Vector Machine*

RN Rede Neuronal

RLE *Run-Length Encoding*

TIC Tecnologias de Informação e Comunicação

VN Verdadeiros Negativos

VP Verdadeiros Positivos

Capítulo 1

Introdução

Nos dias de hoje, as plataformas *online* têm uma presença cada vez maior na educação dos estudantes portugueses. A implementação das chamadas Tecnologias de Informação e Comunicação (TIC), começou realmente após a aprovação do Despacho n.º 116/2005, Série II de 20 de junho de 2005, onde o Gabinete de Informática (GI) passa a ter como principal missão apoiar os utilizadores no uso corrente das TIC (Diário da República nº 116/2005 Série II, 2005). Juntamente com outras medidas definidas nesse documento, pretende-se a massificação do uso social das TIC.

Como resultado da implementação das TIC, os professores tiveram de adaptar o seu método de ensino, fazendo a transição do uso de recursos físicos para os digitais. Com esta mudança, os estudantes passaram a ser avaliados usando plataformas *online*. O Moodle é um exemplo de uma dessas plataformas.

A possibilidade de prever as classificações finais dos estudantes, particularmente quando é feito numa fase inicial do ano letivo, é uma preocupação recente da comunidade académica, uma vez que pode ser usado para fornecer recomendações importantes aos estudantes. Essas recomendações são baseadas em padrões de atividade estudados anteriormente. Como tal, as investigações recentemente feitas na área de análise de aprendizagem têm explorado diferentes caminhos para chegar a esse objetivo, tais como: o uso de *Support Vector Machine*s (SVMs) (Venkat et al., 2018; Liao et al., 2019), *naïve bayes* (Felix et al., 2019) e árvores de decisão (Figueira, 2017). O uso de plataformas *online*, que são capazes de manter um registo de todas as atividades que nelas foram realizadas, proporcionam dados para a realização de estudos sobre padrões de atividade tendo como objetivo final a previsão de classificações.

1.1 Uso do Moodle

O Moodle é um *Learning Management System* (LMS), que serve como plataforma de *e-learning*, desenvolvida por Martin Dougiamas, em Perth na Austrália, e foi lançada a 20 de agosto de 2002 (Moodle, 2019). A plataforma tem maior adesão nas universidades, mas com o passar dos anos

tem-se verificado cada vez mais a integração do Moodle no ensino básico e secundário, devido às características que a plataforma possui. As universidades, comunidades, escolas e professores servem-se dela para comunicar e transmitir informação às suas comunidades educativas (Pimentel, 2009). O uso do Moodle é atrativo não só pelo facto de ser fácil de utilizar, mas também pela sua eficácia e baixo custo. Do ponto de vista de programação, o Moodle é uma plataforma *open-source*, o que permite a vários programadores contribuir com *feedback* e novas características. O que por sua vez, faz com que o número de utilizadores continue a aumentar ao longo do tempo e os benefícios da sua utilização cheguem a mais estudantes, apesar dos programadores não estarem diretamente associados à empresa MoodleHQ, que é responsável pelo desenvolvimento e manutenção da plataforma.

Os professores podem usar o Moodle para fazer o *upload* de ficheiros, definir um calendário de atividades, marcar trabalhos de casa que têm de ser entregues usando a plataforma, realizar testes cuja correção é feita de forma automática, entre outras. As atividades do Moodle podem ser classificadas como individuais (ex: testes) ou cooperativas, que incluem trabalhos de grupo e fóruns de discussão onde os estudantes discutem temas que eles próprios propõem (Lopes, 2011).

Fazendo uso das suas características, existem alguns benefícios diretos para os professores, tais como:

- Promover a discussão entre estudantes por meio de um fórum;
- Disponibilizar conteúdos como ficheiros e calendário da unidade curricular para todos os estudantes, mesmo aqueles cuja assiduidade é baixa;
- Autocorreção dos testes, liberta muito tempo útil para o docente, podendo aplicar noutras atividades académicas, que não correções;
- A existência de um repositório comum a todos, onde o professor pode receber vários ficheiros submetidos pelos seus estudantes como parte de trabalhos de submissão.

Apesar de todas as vantagens que existem, os utilizadores apontam algumas desvantagens que dizem respeito a problemas técnicos do LMS, à necessidade de se dedicar à preparação dos ambientes (imagem da unidade curricular, calendarização das ações) dos recursos, que consomem muito tempo (Pimentel, 2009).

1.2 Motivação

Durante a última década, em vários países europeus, o rácio de estudantes por professor tem vindo a aumentar de forma contínua e sistemática (Figueira, 2015). Em Portugal, esta situação é ainda mais evidente e aconteceu devido à redução do número de professores e ao aumento do número de estudantes por turma (Direção-Geral de Estatísticas da Educação e Ciência, 2018). Como tal, o número de estudantes que têm de ser diretamente avaliados e ensinados por um só

professor aumentou. Portanto, é esperado que um professor faça mais, usando menos recursos. É assim que plataformas de aprendizagem *online* como o Moodle, que permite aos professores realizar mais tarefas de forma mais fácil e rápida, se tornam muito úteis.

O uso generalizado do Moodle como uma ferramenta de auxílio à aprendizagem, torna possível o surgimento de novos estudos que têm como base o comportamento dos estudantes, devido à forma como o Moodle cria e gere os seus ficheiros de *log*.

Estes ficheiros de *log* são uma forma de registo automático, feito pelo Moodle, onde são guardadas informações sobre o uso da plataforma, quais os recursos usados e a sequência de acesso aos recursos pelos utilizadores. Todas as atividades realizadas nesta plataforma são registadas e mantidas nos ficheiros de *log*, com a indicação do dia e hora em que se iniciaram.

É fácil entender como a existência deste tipo de informação permite a descoberta e o estudo de novos padrões de atividades para cada estudante, devido ao elevado detalhe da informação disponível. É a partir desta informação que podem ser criados modelos que analisam todas as atividades de um dado estudante e determinam a probabilidade dele ter uma boa, razoável ou má classificação final.

Devido à natureza *open-source* do Moodle, seria possível integrar o modelo proposto na plataforma, usado por exemplo uma interface fácil de usar e aceder. Esta implementação iria afetar um vasto número de estudantes, que passariam a poder consultar qual seria a previsão da sua classificação final. Por outro lado, aqueles estudantes em risco de reprovar, quando confrontados com as consequências das suas ações, iriam mudar os seus comportamentos de maneira a melhorar a sua classificação final. Já o docente, que também teria acesso a esta informação, poderia usá-la para determinar quais os estudantes que necessitam mais do seu apoio. Ambas as perspetivas têm como potencial consequência o aumento da taxa de sucesso em todas as unidades curriculares, que usem o Moodle como plataforma de aprendizagem.

1.3 Objetivos de Investigação

O principal objetivo deste estudo, é determinar se existe uma forma de relacionar padrões de acesso *online* com os resultados da aprendizagem, obtidos por estudantes do ensino superior. Para atingir esse objetivo, é proposta a criação de um programa capaz de identificar padrões de atividade de cada estudante e fazer comparações com os restantes, usar os padrões de atividade *online* para prever a nota final de cada estudante, identificar casos de fraude académica e permitir a consulta do percurso *online* de um estudante, por ele mesmo e/ou pelo seu docente. Assim, existem questões de investigação que devem ser respondidas, e que são listadas a seguir:

Q1) É possível prever o resultado final de cada estudante com base, nas interações entre este e a plataforma?

Q2) É possível saber-se com antecedência, antes de terminar o semestre, se um estudante vai reprovar à unidade curricular?

Q3) Que tipo de modelo preditivo poderá ser usado?

Q4) Existe alguma forma de calcular quanto tempo é que o estudante passou numa atividade do Moodle?

Q5) Será possível criar uma metodologia capaz de comparar o percurso académico de dois estudantes e no final atribuir um valor a essa comparação?

1.4 Metodologia

Inicialmente, antes de qualquer tipo de implementação, foi realizado um estudo da literatura disponível de maneira a determinar que estratégias deveriam ser adotadas e quais as abordagens a evitar. Este estudo permite também a clarificação e uniformização de conceitos, que levam a um maior entendimento do estado atual destes conceitos e metodologias na comunidade científica.

Após concluída esta fase, foi realizado um estudo exploratório detalhado dos dados, a serem usados. Terminada a exploração dos dados a próxima etapa foi a de eliminação de ruído, onde todas as entradas que não são consideradas relevantes foram eliminadas, foi feita a extração e separação de informação que, se encontra presente nos dados originais. A eliminação, extração e classificação dos dados foram realizadas, com uso de vários programas desenvolvidos e executados sequencialmente, criados usando a linguagem de programação Python. A etapa seguinte foi a de criação do modelo, baseado no uso de um algoritmo de Machine Learning, que é capaz de analisar os comportamentos, extrair informação baseada nas atividades e no final prever qual a classificação final. A fase seguinte foi a de testes onde, foram feitos vários testes com o objetivo de determinar a precisão e exatidão do modelo criado, de maneira a garantir que o modelo é eficiente e capaz de atingir os objetivos pretendidos.

De forma a atingir o objetivo deste estudo, o trabalho realizado foi dividido em várias fases.

1) Estudo do estado da arte;

2) Análise e extração de informação dos utilizadores do Moodle;

3) Preparação dos dados;

4) Identificação de padrões de atividade;

5) Associação de padrões de atividade a uma classificação final;

6) Estabelecer comparações entre vários estudantes e os seus diferentes padrões de atividade;

7) Definir quais as variáveis independentes e qual a variável objetivo;

8) Criar conjuntos de treino e de teste;

9) Aplicar um algoritmo de Machine Learning;

10) Avaliar as previsões feitas pelo algoritmo;

11) Melhorias do modelo;

12) Escrita da tese.

1.5 Organização da Tese

Este documento foi dividido em seis capítulos. Introdução capítulo 1, onde é feita uma breve explicação de qual o problema abordado e que medidas foram tomadas para o resolver. Conceitos capítulo 2, neste capítulo são explicados todos os diferentes conceitos que foram usados e/ou explorados na criação do modelo. Trabalho relacionado capítulo 3, onde foram verificados e consultados os artigos que foram publicados e que de alguma forma se encontram relacionados com o problema que se pretende resolver e fez-se o estudo do estado da arte. O capítulo 4, contém uma descrição do todo o trabalho efetuado, desde a fase inicial de recolha de dados até a final de obtenção de resultados. O capítulo 5, explora todos os resultados obtidos e faz a sua análise. Por último, a conclusão, o resumo das contribuições e trabalho futuro no capítulo 6.

Capítulo 2

Conceitos

Neste capítulo, são explorados vários conceitos relacionados com termos e técnicas usadas, que auxiliam a previsão de classificações finais e a avaliação dessas previsões, tendo como base os padrões de atividade dos estudantes.

2.1 Balanceamento de Classes

A maioria dos *datasets* sofrem de desequilíbrio de classes, *class imbalance*, quando o número de observações em cada classe é desigual. Neste contexto, muitos algoritmos de Machine Learning têm baixa precisão preditiva para a classe que não é frequente. Uma solução para resolver este desequilíbrio é utilizar o *oversampling* e o *undersampling*. Estas técnicas são usadas para ajustar a distribuição de classe de um *dataset*, restaurando assim o equilíbrio.

O *oversampling* envolve o aumento do conjunto de treino com várias cópias de algumas das classes minoritárias e é um processo que pode ser realizado mais do que uma vez. Em vez de duplicar todas as amostras na classe minoritária, algumas delas são escolhidas aleatoriamente com substituição (Ling e Li, 1998).

O *undersampling* remove aleatoriamente amostras da classe maioritária sem substituição. Esta é uma das principais técnicas usadas para aliviar o desequilíbrio do *dataset*, no entanto, pode potencialmente descartar amostras úteis ou importantes (Fernández et al., 2018).

Estas duas técnicas são aplicadas a um conjunto de treino até que todas as classes apresentem o mesmo número de amostras e se encontrem representadas da mesma forma. Depois de aplicadas estas duas técnicas a um conjunto de treino, é garantido que todas as classes apresentam o mesmo número de amostras e estão representadas da mesma forma.

2.2 Classificação

Para definir se um modelo atingiu bons resultados quando aplicado um algoritmo de Machine Learning, existem conceitos de estatística que servem para classificar a qualidade preditiva do modelo (Powers, 2011).

2.2.1 Matriz de Confusão

Uma matriz de confusão é uma tabela geralmente usada para descrever o desempenho de um modelo de classificação em conjunto com o *dataset* para os quais os valores verdadeiros são conhecidos. A matriz descreve uma saída negativa vs. positiva. Estes dois resultados são as "classes" de cada exemplo. Como existem apenas duas classes, o modelo usado para gerar a matriz de confusão pode ser descrito como um classificador binário.

De maneira a interpretar melhor esta tabela pode-se pensar em termos, como: verdadeiro negativo, falso positivo, falso negativo e verdadeiro positivo.

Tabela 2.1: Matriz de confusão.

	Negativo (Previsto)	Positivo (Previsto)
Negativo (Real)	Verdadeiro Negativo (VN)	Falso Positivo (FP)
Positivo (Real)	Falso Negativo (FN)	Verdadeiro Positivo (VP)

2.2.2 *Accuracy*

A *accuracy* (AC) representa quantas vezes é que o modelo consegue prever corretamente uma classificação. Indica quão perto os valores obtidos estão do valor real, serve para indicar se um modelo está a ser treinado corretamente e também qual o seu desempenho.

Este valor é calculado dividindo o número de valores verdadeiros obtidos pelo número total de valores.

$$AC = \frac{VP + VN}{VP + FP + VN + FN} \tag{2.1}$$

2.2.3 *F-Score*

O *F-Score* é um indicador uni-dimensional, que atualmente representa uma métrica importante na avaliação do desempenho de um sistema de *Natural Language Processing* (NLP) ou de *Information Retrieval* (IR) (Huang et al., 2015). Este *score* é representado com valores entre 0 e 1, sendo que quanto mais elevado for o *F-Score*, maior é o número de verdadeiros positivos e verdadeiros negativos e, logicamente, menor é o número de falsos positivos e falsos negativos. Um elevado

valor de *F-Score* indica que o modelo é capaz de identificar corretamente os valores verdadeiros e não é afetado por valores falsos.

O *F-Score* é calculado utilizando *precision* (PC) e *recall* (RC).

$$F1 = 2 * \frac{(PR * RC)}{PR + RC} \tag{2.2}$$

2.2.4 *Precision*

Quando o modelo é capaz de prever um resultado positivo, isto é, indica quantas vezes é que o resultado obtido é o correto.

Calcula-se dividindo o número de verdadeiros positivos pelo número total de positivos.

$$PR = \frac{VP}{VP + FP} \tag{2.3}$$

2.2.5 *Recall*

Recall (RC) ou "sensitividade" indica qual a percentagem de verdadeiros positivos que foram corretamente identificados como sendo verdadeiros positivos. Ao contrário da *accuracy* que identifica quantas vezes é que o modelo é capaz de obter valores verdadeiros positivos e negativos. O *recall* apenas indica o número de vezes que o modelo consegue obter valores verdadeiros positivos.

Resulta da divisão do número de valores positivos pela soma de falsos negativos com verdadeiros positivos.

$$RC = \frac{VP}{FN + VP} \tag{2.4}$$

2.3 *Data Mining*

A extração manual de padrões usando dados já acontece há séculos. *Data Mining* surge da estatística, onde eram usados métodos de identificação de padrões em dados. A expansão e evolução dos computadores, aliadas à sua adoção global, tornou possível a recolha, armazenamento e extração de grandes quantidades de dados.

À medida que os conjuntos de dados aumentaram em tamanho e complexidade, a análise direta dos dados aumentou com seu o processamento indireto e automatizado. Certas descobertas na área de ciências de computação ajudaram a tornar o processo de extração e estudo dos dados

mais rápido, como as redes neuronais, análise de *clusters*, algoritmos genéticos, *decision trees* e *Support Vector Machine* (SVM).

Data mining é o processo de aplicar estes métodos com a intenção de descobrir padrões ocultos em grandes conjuntos de dados (Kantardzic, 2011).

Inicialmente foi feito um estudo sobre o que é *data mining* quando aplicado a este contexto e quais são as técnicas padrão para processar este tipo de dados.

Wirth (2000) defende que *data mining* precisa de uma abordagem padrão que ajuda: a traduzir problemas técnicos ou formais em tarefas de *data mining*; sugere transformações de dados apropriadas; técnicas de *data mining* e fornece meios para avaliar a eficácia dos resultados, bem como documentar a experiência.

2.4 *Data Preparation* e *Data Cleaning*

A preparação de dados é uma das fases fundamentais da análise de dados (Zhang et al., 2003). Consiste na alteração ou pré-processamento dos dados no seu formato original, para um formato que permita a sua análise. Este processo consiste na recolha, organização, limpeza e consolidação dos dados num único ficheiro, usado para análise. Existem tarefas de pré-processamento que têm de ser realizadas (previamente): *data fusion* e *data cleaning* antes de aplicar os dados recolhidos nos ficheiros de *log* final a algoritmos de *data mining* (Cooley et al., 1999).

A limpeza de dados, também conhecida como *data cleansing* ou *data scrubbing*, é um dos processos que constitui a preparação dos dados. O problema da qualidade dos dados recolhidos em bases de dados é algo preocupante, devido à existência de dados redundantes. A limpeza dos dados surge como uma solução a este problema (Rahm et al., 2000). A limpeza corresponde ao processo de detetar e eliminar do *dataset* as observações que são consideradas irrelevantes para o estudo ou que se encontrem incompletas (o chamado "ruído"), garantindo, assim, que o conjunto de dados final contém apenas a informação que é relevante e que está completa para ser usada no estudo.

2.5 Distância Euclidiana

A distância Euclidiana, $d(p, q)$, representa a distância entre dois pontos (p e q) em linha reta, num espaço Euclidiano constituído por n dimensões ortogonais onde $n \in \{1, 2, ..., n\}$ (Maurer et al., 2003; Wang e Tan, 2013). A $d(p, q)$ pode ser calculada através da raiz quadrada do somatório a n dimensões do quadrado das subtrações das coordenadas cartesianas de cada um dos pontos, assumindo que $p = (p_1, p_2, ..., p_n)$ e $q = (q_1, q_2, ..., q_n)$ (Breu et al., 1995).

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{2.5}$$

2.6 Machine Learning

2.6.1 Introdução

Machine Learning é uma aplicação de Inteligência Artificial (IA), que fornece aos sistemas a capacidade de aprender e melhorar automaticamente a partir da experiência, sem serem explicitamente programados. O foco deste tipo de aplicação é criar sistemas que sejam capazes de aceder a dados e usá-los para "aprenderem" por si mesmos (Pedregosa et al., 2011). Devido a estas características, Machine Learning pode ser descrito como sendo o estudo científico de algoritmos e modelos estatísticos que os sistemas de computador usam para executar, com elevada eficácia, uma tarefa específica, sem usar instruções explícitas pré-programadas, baseando-se em padrões e inferência.

O processo de "aprendizagem" começa com as observações ou dados reunidos, que são usados como exemplos, de maneira a permitir a procura de padrões nos dados e garantir que o sistema é capaz de tomar melhores decisões no futuro, com base nos exemplos inicialmente recolhidos. O objetivo principal é permitir que os computadores aprendam automaticamente, sem intervenção humana ou assistência a ajustar as ações de acordo com a necessidade (Bishop, 2006).

2.6.2 *Decision Trees*

Uma árvore de decisão é uma ferramenta de suporte à decisão, usada tanto em *data mining* como em Machine Learning. Este algoritmo usa um gráfico ou modelo semelhante a uma árvore, onde as divisões entre os ramos indicam quais são as regras de decisão ou *decision rules* que foram aplicadas de forma a obter o subconjunto de dados. Trata-se de uma forma de representar um modelo de decisões (Magerman, 1995). Uma árvore de decisão é classificada como um procedimento de classificação que particiona recursivamente um conjunto de dados em subdivisões menores, com base num conjunto de regras de decisão definidas em cada ramo (ou nó) da árvore. Os nós são criados de forma a minimizar o mais rapidamente possível a entropia. A árvore é composta por três tipos de nós. Contém um único nó raiz (formado a partir de todos os dados), um conjunto de nós internos (divisões) e um conjunto de nós terminais (folhas). A figura 2.1 demonstra este tipo de representação.

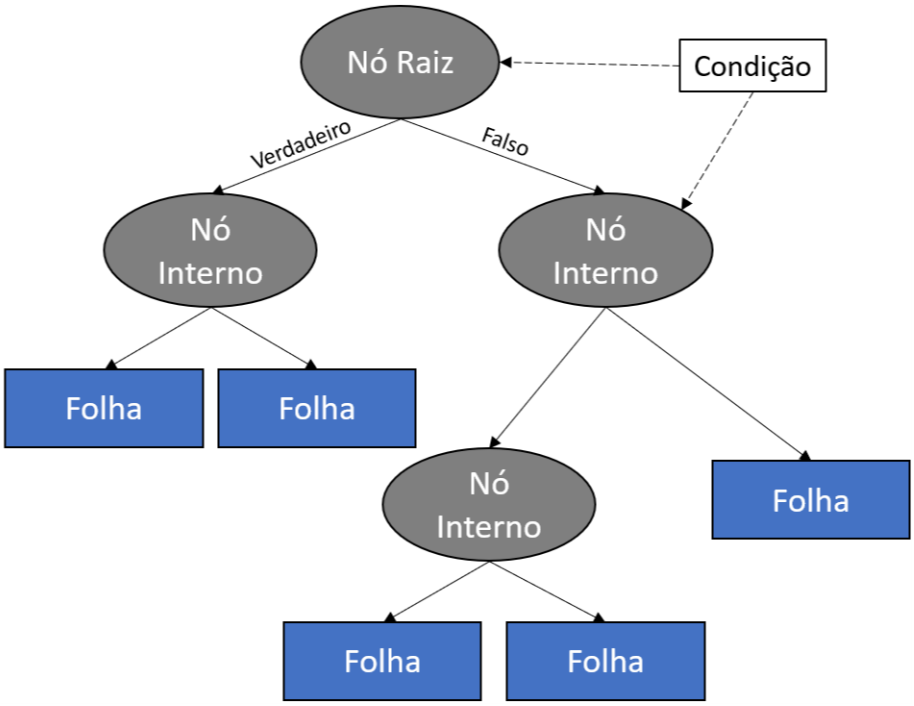


Figura 2.1: Representação de uma árvore de decisão (Najm et al., 2019).

Cada nó numa árvore de decisão binária contém apenas um nó pai e dois nós descendentes, podendo estes ser nós internos ou nós terminais (Friedl e Brodley, 1997; Safavian e Landgrebe, 1991). Por sua vez, cada um dos nós descendentes, pode ou não originar dois nós por cada nível da árvore de decisão.

Seguindo esta estrutura, um conjunto de dados é classificado sequencialmente e subdividido de acordo com o conjunto de regras de decisão definido pelo algoritmo, e uma classificação ou classe é atribuída a cada uma das observações, de acordo com o nó da folha, em que a observação se encontra inserida.

Este algoritmo possui várias vantagens quando comparado com algoritmos de classificação supervisionada. O algoritmo utiliza divisão binária recursiva, tenta decidir quais os recursos a escolher e quais as condições que devem ser usadas para dividir os nós, (*split*), além de saber quando é que a árvore deve ser parada. Em cada uma das divisões, segundo as regras de divisão, o algoritmo calcula qual será a *accuracy* de cada divisão. É feita uma comparação e a divisão que apresenta a maior *accuracy* é escolhida. O algoritmo é recursivo porque os conjuntos obtidos podem depois voltar a ser divididos em sub-conjuntos, utilizando-se o mesmo processo. Devido a este processo, o algoritmo pode ser classificado como *greedy*.

2.7 Modelos Preditivos

A modelação preditiva é um processo que usa *data mining* e probabilidade para prever resultados. Cada modelo é constituído por diversos indicadores, que são variáveis e que provavelmente influenciam os resultados. Após a recolha de dados que são utilizados em conjunto com variáveis independentes, o modelo preditivo é formulado. O modelo pode ser constituído por uma simples equação linear ou pode ser uma Rede Neuronal (RN) complexa. Este tipo de modelação é usado, em várias áreas desde previsões meteorológicas, consultoria financeira, deteção de fraude, entre outras (Khum e Johnson, 2016).

2.8 *Run-Length Encoding*

Existem vários métodos de compressão diferentes, entre estes destacam-se dois tipos: *lossless* e *lossy*. O tipo *lossless* garante que os dados originais não são perdidos quando é feita a compressão, enquanto que, usando o método *lossy*, alguma da informação é perdida e não pode ser recuperada quando a informação volta ao formato original. Dependendo da situação, há momentos em que a compressão com perdas (*lossy*) é preferível por exemplo, quando o objetivo é obter um ficheiro com tamanho menor ou se a perda de dados não for percetível. Nos casos em que qualquer tipo de perda de dados é inaceitável, a compactação sem perdas (*lossless*) é preferida, embora o tamanho do arquivo seja significativamente maior.

O *Run-Length Encoding* (RLE) é classificado como um método de compressão *lossless*, onde "conjuntos" (elementos de dados consecutivos) são substituídos pelo número de ocorrências e o respetivo valor de dados (Xu et al., 2004).

RLE também pode ser usado para transformar várias *strings* separadas, numa única *string*. É mantido o nível de detalhe da informação, mas num formato mais simples de usar (Hinds et al., 1990).

Capítulo 3

Trabalho Relacionado

3.1 Publicações Semelhantes

O estado atual dos estudos realizados sobre *Educational Data Mining* (EDM), que é a vertente de *data mining* focada na educação, foi investigado por Romero e Ventura (2010). Este estudo analisa o número de *papers* publicados entre 1993 e 2009, cujo tema se encontra relacionado com EDM. Destacam-se o aumento de publicações e o interesse no tema, a partir do ano 2000. Foram analisadas 236, publicações, cada uma foi classificada como pertencendo a uma de oito categorias: diferentes, *traditional education*, *web-based education/e-learning*, *learning management systems*, *intelligent tutoring systems*, *adaptive educational systems*, *tests/questionnaires*, *texts/contents* e outros. Conclui-se que o interesse no uso de *data mining* aplicado ao contexto da educação, tem vindo a aumentar bastante nos últimos anos. Aumentou também o interesse em tentar integrar estes sistemas, *Learning Management System* (LMS), na educação.

3.2 Utilização das Tecnologias de Informação e Comunicação (TIC)

Com o passar dos anos, o Governo Português tem promovido a integração das chamadas TIC no ensino português (Lisbôa et al., 2009). Com a implementação desta medida e outras parecidas, foram abertas as portas para a criação de novos estudos e descobertas, porque muitas das plataformas a serem integradas mantêm um registo ativo de todas as atividades a realizar. Esta quantidade vasta de informação disponível para ser analisada era, anteriormente, bastante difícil de estudar.

Logicamente foram desenvolvidas novas metodologias, para analisar e classificar estes dados, muitas delas fortemente relacionadas com técnicas usadas em *data mining*.

O estudo realizado por Dias et al. (2016) conclui que a plataforma Moodle, devido às suas características e funcionalidades, desperta o interesse dos professores em integrar esta plataforma

ou outras parecidas no seu método de ensino. Portanto, pode ser dito que, a adoção destas novas tecnologias para melhorar e simplificar a qualidade do ensino português vai aumentar com o passar dos anos (Fernandes, 2008).

O Moodle, como ferramenta de ensino, pode ser aplicada em dois contextos distintos, o do ensino superior e do ensino básico/secundário. Pimentel (2009) diz que, ao nível do ensino básico, constituído pelo primeiro ciclo (1º, 2º, 3º e 4º ano), pelo segundo ciclo (5º e 6º ano) e terceiro ciclo (7º, 8º e 9º ano), são os professores do terceiro ciclo que mais frequentemente utilizam a plataforma. Cada um dos professores e educadores entrevistados nesse estudo, falam das vantagens associadas à utilização do Moodle, sobretudo como repositório de trabalhos e de informações importantes a partilhar com outros professores ou com os estudantes, uma vez que a facilidade de acesso a conteúdos e serviços é grande (Pimentel, 2009). Já no ensino superior, o uso deste tipo de ferramentas é generalizado. Existem várias unidades curriculares que integram o Moodle no seu método de aprendizagem, seja como forma de realizar testes, entrega de trabalhos e disponibilização de conteúdos, fórum e notícias (Costa et al., 2012).

Foi concluído por Dias et al. (2016) e por Pimentel (2009), que uma das principais razões para a fraca adoção desta ferramenta, por parte dos professores, nos "baixos" escalões de ensino é o facto de requer algum conhecimento prévio ou formação, para que estes possam tirar partido de todas as funcionalidades da plataforma. O aumento da carga horária a que os professores estão sujeitos atualmente, faz com que muitos sintam que a adoção de uma nova ferramenta seja algo desnecessário.

Relativamente à taxa de utilização do Moodle, Duarte e Gomes (2011) diz-nos que é no ensino superior onde a plataforma tem maior uso e, como tal, é neste escalão onde a recolha de dados é maior. Portanto, este estudo tem como foco este escalão da educação e fará uso dos dados recolhidos nele, para criar um modelo preditivo que servirá de alerta tanto para os estudantes como para o docente. Já Martins et al. (2019), questionam os estudantes e professores do ensino superior sobre a perceção que estes têm da plataforma. Este estudo revelou que a maioria, (89%), dos estudantes considerou a plataforma como sendo maioritariamente um repositório de ficheiros enquanto que apenas 36% dos professores, concordam com essa declaração. Estes resultados mostram que apesar da plataforma ser usada quer pelos professores como pelos estudantes, existem diferentes interpretações e abordagens quanto ao seu uso.

Escobar-Rodriguez e Monge-Lozano (2012), estudam a possível integração Moodle no processo académico de aprendizagem de estudantes do curso de economia do ensino superior. Neste artigo é descrita uma ferramenta, de que forma pode ser integrada no método de educação e qual a perceção que os estudantes têm sobre a sua utilidade. É concluído que, após demonstrar todas as possibilidades e capacidades da plataforma, os estudantes encontram-se muito recetivos à sua utilização como uma plataforma de auxílio ao ensino.

3.3 Estudos Exploratórios

3.3.1 Quais as Atividades/*Features* a Usar?

Tendo como base as conclusões retiradas do estudo de Figueira (2016), após uma análise dos *logs* do Moodle e a criação de uma árvore de decisão, são obtidas conclusões úteis, como quais as *features* que têm maior influência na nota final do estudante. Estas *features* conseguem também demonstrar o carácter e preparação do estudante, através do número de acessos à atividade de Workshop, que é considerado um bom indicador. Mas quando, é comparado com o número de visualizações, é fácil de entender que os estudantes, além de demonstrarem uma baixa produtividade, também demonstram insegurança e medo de cometer erros.

Assim, entre outras conclusões ao analisar o estudo Figueira (2016), consegue-se perceber quais as atividades que mais influenciam a nota e quais as ligações que podemos estabelecer entre elas, de forma a delimitar um "perfil" de utilizador.

3.3.2 Eficácia de Algoritmos de Machine Learning

Um estudo recente de Đurđević Babić (2017) tenta fazer uma comparação entre a eficácia de diversos métodos de Machine Learning, para prever a motivação dos estudantes, de maneira a determinar qual desses métodos é o melhor a efetuar previsões. Segundo os padrões e critérios de classificação usados, após comparar Rede Neuronal (RN), *classification tree* e *Support Vector Machine*s (SVMs), foi concluído que as RNs são de facto as que apresentam uma precisão maior quando se trata de prever classificações, com um valor de cerca de 76.92%. Este estudo é interessante porque além de demonstrar o desempenho de cada um dos modelos adotados, também fornece informação e alguns detalhes de como deve ser feita a classificação de padrões e de critérios.

A criação de um sistema de *e-learning* inteligente capaz de prever as notas finais dos estudantes é o objetivo de Simjanoska et al. (2014), que descreve detalhadamente como deve funcionar um sistema desse tipo, que tipo de interações ocorrem e de que forma deve ser aplicado. Este estudo foca-se em introduzir *Intelligent Grading Agent* (IGA) numa plataforma de *e-learning*, fazendo uso de dois tipos de SVMs. O estudo prova a eficácia deste método com valores de precisão compreendidos entre 98.18% e 100%, quando aplicados aos casos de teste usados pelos autores. Os resultados mostram o potencial do uso de SVMs e também a importância que estes resultados podem ter.

Já o estudo realizado por Venkat et al. (2018), conclui que, analisando vários algoritmos de Machine Learning, como *naïve bayes*, *decision tree* e SVMs, o último é o que apresenta a melhor *accuracy* quando comparado com os restantes ao longo de três testes.

Mais recentemente, Liao et al. (2019) apresentou um estudo onde é descrito um sistema de alerta para estudantes. Este sistema usa informação recolhida por um *clicker* e uma SVM, para

alertar, os estudantes, sobre comportamentos negativos que iram levar a uma reprovação na unidade curricular.

Apesar destes estudos claramente classificarem as árvores de decisão como não sendo o algoritmo ideal a utilizar na realização de previsões de notas, este estudo utilizou esse mesmo algoritmo para poder explicar em que se baseia a classificação atribuída.

3.3.3 Como Prever Classificações?

Um dos métodos para determinar a classificação final dos estudantes foi apresentado por Figueira (2015), onde após a identificação dos padrões de atividades dos estudantes, são escolhidos três vetores de dados de teste. Estes são escolhidos manualmente como sementes para a criação de um determinado *cluster*: notas boas, médias ou más. De seguida, é gerada uma *centroid* daquele *cluster*. O processo é repetido para cada um dos *clusters*. No final, o modelo é capaz de receber um determinado padrão de atividades e produz como resultado uma classificação da nota final obtida: se é má, média ou boa.

Calvo-Flores et al. (2006), propõem o uso de uma RN para fazer previsões, utilizando o Moodle como ferramenta de recolha de dados. Foram analisados 240 casos e foi extraída informação que permitiu a criação de diversas variáveis, que auxiliam o modelo preditivo. Este modelo obteve uma elevada taxa de sucesso, sendo capaz de identificar corretamente os estudantes que se encontram em risco de não conseguir uma nota positiva, na unidade curricular.

Outro estudo propõe a criação de uma aplicação capaz de determinar o estado atual de estudantes e turmas, em termos do seu conhecimento e também de que forma deve ser feita essa avaliação (Fonseca et al., 2016). Este estudo, para além de explicar e clarificar quais as características bases que um sistema desta natureza deve ter, também explica quais as limitações/problemas que podem advir da aplicação de um sistema deste tipo e indica de que forma devem ser analisados os comportamentos dos estudantes.

Vahldick et al. (2017), determina de que forma a implementação de um jogo nas aulas pode ajudar e melhorar a maneira como estudantes que não possuem nenhum conhecimento inicial sobre programação, aprendem a ultrapassar as barreiras iniciais de programar. Nesse estudo é usado *Learning Analytics* (LA) e um conjunto de nove regras para, consoante o tempo passado pelos estudantes num determinado nível, determinar se o seu nível de conhecimento é bom, mau ou excelente. Foi concluído que o uso de uma ferramenta deste tipo é importante para que o professor entenda qual o conhecimento adquirido pelos seus estudantes e também permite ao professor dar mais atenção e maior ajuda a estudantes cujo nível de conhecimento é baixo.

3.3.4 Cuidados a Ter

Devido ao seu amplo uso na educação, alguns estudos (Gašević et al., 2016; Conijn et al., 2017) tentaram analisar os dados recolhidos pelo LMS para prever as notas finais dos estudantes e

tentar estabelecer uma relação entre o uso de um LMS e o desempenho académico dos estudantes. Certos estudos usam “preditores diretos” (Conijn et al., 2017), informações que são extraídas diretamente dos ficheiros de *log*, por exemplo o tempo total *online*, número de cliques e diferentes módulos visitados/usados (fóruns, questionários, etc). O conceito de "preditores diretos" foi usado neste estudo.

Os estudos (Conijn et al., 2017) e (Gašević et al., 2016), fazem uso de várias unidades curriculares, como forma de recolher informações para serem usadas no conjunto de dados. No entanto, o uso de dados recolhidos de unidades curriculares diferentes pode criar um problema, uma vez que não pode ser assegurado que múltiplas unidades curriculares tenham a mesma abordagem e atribuem a mesma relevância ao uso do LMS no processo de aprendizagem.

Naturalmente, podemos supor que professores diferentes têm diferentes métodos de ensino, atribuem diferentes importâncias ao uso de ambientes *online* e têm metodologias de avaliação diferentes. Portanto, pode-se concluir que como os dados recolhidos, por esses estudos, pertencem a várias unidades curriculares diferentes, é difícil manter um grau de variação mínimo, no ambiente em que os dados foram recolhidos. Além disso, os estudos mencionados concluíram que a portabilidade do modelo preditivo entre diferentes unidades curriculares é limitada, o que significa que existem muitas diferenças no uso, dependência e importância que cada unidade curricular atribui ao LMS. Por outro lado, os estudos concluíram que esse tipo de modelos preditivos podem ser aplicados com sucesso a um único curso. Quanto menor o número de estruturas do curso (avaliação e metodologia), a partir da qual os dados são recolhidos, menor é a variação da estrutura da unidade curricular e, portanto, é mais fácil prever corretamente uma nota.

Garantir um pequeno grau de variação foi uma das principais preocupações deste estudo. Utilizamos dados recolhidos de uma única unidade curricular, abrangendo três anos letivos, onde o professor que lecionava a unidade curricular foi sempre o mesmo e, como tal, o método de avaliação também foi o mesmo. Consequentemente, podemos supor que a mesma importância foi dada a cada um dos componentes da unidade curricular.

3.3.5 Sistemas Semelhantes

De forma a determinar a importância da atividade de fórum, o estudo de López et al. (2012) tenta definir de que forma a participação no fórum influencia a nota final obtida pelos estudantes. Esse estudo utiliza dados recolhidos de 114 estudantes e as 1.014 interações entre eles. É usado um algoritmo de *clustering* como modelo preditivo. O estudo conclui que a atividade do fórum pode ser considerada uma boa variável independente, pois tem um grande impacto na nota final e conclui também que a classificação utilizando *clustering* apresenta uma *accuracy* elevada, o que indica que o modelo é eficaz.

O estudo realizado por Felix et al. (2019), propõe a criação de um sistema chamado MoodlePredicta que é capaz de prever resultados académicos. Ao longo do artigo é descrita a

ferramenta, divida em duas componentes: a visual e a de previsão. A componente visual permite ao estudante analisar todo o seu percurso. A componente de previsão utiliza *naïve bayes* para determinar a nota final. Os dados foram recolhidos de 13 unidades curriculares, tendo um número total de 1.307 estudantes. Este estudo descreve de que forma uma ferramenta deste tipo, deve estar organizada, de maneira a ter uma *interface* simples de usar e entender pelo utilizador.

Iglesias-Pradas et al. (2015) realizaram um estudo, onde utilizam *learning analytics* para definir preditores, que estejam relacionados com *teamwork* e o empenho dos estudantes. O estudo analisa as entradas do Moodle para identificar os comportamentos que determinam, para aquele estudante, qual o seu nível de empenho e a sua capacidade de trabalhar em equipa. Para tal, avaliam as interações que existem entre os estudantes. Os resultados obtidos são contraintuitivos, uma vez que não apresentam qualquer tipo de relação entre o uso do LMS e o nível de competências que foi adquirido.

Figueira (2016), propõe e discute a possibilidade de criar um modelo capaz de prever classificações, no final do semestre, usando informação extraída das observações do Moodle (tempo passado em cada uma das atividades). Neste artigo são analisadas todas as atividades e determinada qual a influência que cada uma tem na nota dos estudantes. Essa informação é depois usada em conjunto com uma *decission tree*, para fazer previsões sobre as notas finais dos estudantes. Este estudo é útil na medida em que serve para identificar quais as atividades com maior peso na nota final e, como tal, ajuda a determinar preditores relacionados com essas atividades.

3.4 Conclusões Retiradas

A análise e estudo de publicações relacionadas com o tema desta dissertação serviu para determinar o que foi feito anteriormente, identificar quais os erros que devem ser evitados e retirar novas conclusões/ideias que podem ser implementados ao longo da dissertação.

Com essas conclusões surgiu a ideia de implementar uma metodologia, que seja capaz de prever classificações em qualquer ponto do semestre. Este sistema utiliza os dados recolhidos pelo LMS, Moodle, uma vez que é das plataformas com maior taxa de utilização em Portugal. O escalão de ensino usado na recolha foi o do ensino superior, uma vez que apresenta a maior taxa de adoção desta plataforma. Foi usada apenas uma unidade curricular, esta foi a conclusão obtida após a consulta de vários estudos que faziam uso de mais do que uma unidade curricular, até nove, para a recolha de dados. Após a recolha e tratamento dos dados, são criadas variáveis independentes e uma dependente. Apesar de não ser o algoritmo mais eficiente, foi criada uma árvore de decisão, que juntamente com as variáveis criadas, pode ser aplicada a um conjunto de treino para permitir ao modelo prever classificações. Posteriormente são realizados testes e obtidos resultados.

No capítulo seguinte é explicado o processo de criação do modelo, começando pela estrutura da unidade curricular, criação de novos campos, limpeza e organização dos dados e terminando

com a criação de variáveis independentes, que permitem ao modelo realizar as previsões.

Capítulo 4

Desenho e Desenvolvimento

4.1 Estrutura da Unidade Curricular

4.1.1 Abordagem

Antes de proceder à análise das observações recolhidas, foi necessário entender de que forma é que a unidade curricular se encontra organizada. A informação recolhida é referente à unidade curricular, Comunicação Técnica (DPI1001), que foi usada como caso de estudo. Esta unidade curricular apresenta a particularidade de recorrer ao uso intensivo do Moodle. A plataforma é usada para facilitar o acesso dos estudantes a recursos, tais como conjuntos de slides, testes, *templates*, acesso a entregas de trabalhos e a fóruns onde, os estudantes podem discutir e responder a dúvidas entre si, com a supervisão do docente. O professor também desfruta da capacidade que o Moodle tem de fazer autocorreções de todos os testes, facilitar a entrega de trabalhos e disponibilização de conteúdos.

A unidade curricular tem como principal objetivo preparar os estudantes para a escrita e avaliação de artigos ou relatórios, criação de slides, de apresentações e técnicas para realizar uma apresentação oral em ambiente técnico-científico.

Os estudantes desta unidade curricular, são avaliados em cinco etapas. A *timeline* dos eventos é representada na figura 4.1. As primeiras três etapas são os três testes realizados através do Moodle, que decorrem ao longo do ano letivo. As questões colocadas são baseadas em conceitos, lecionados durante as aulas teóricas, e conteúdos, que se encontram presentes nos materiais de estudo disponibilizados pelo professor.

A próxima etapa é realizada individualmente e corresponde à criação de um artigo. Durante o semestre são dadas *guidelines* sobre como elaborar um artigo e no fim os estudantes têm que submeter um artigo criado individualmente, que pode abordar qualquer tema escolhido. Este artigo é depois avaliado pelo professor, e por um conjunto de estudantes aleatoriamente escolhido, usando a atividade Workshop do Moodle.

A última etapa consiste na avaliação do trabalho de grupo. Cada estudante integra uma equipa de quatro ou cinco estudantes, que fica responsável pela criação de um conjunto de slides sobre um determinado tema. O grupo é responsável por expor o seu tema ao resto da turma por meio de uma apresentação oral. A avaliação do trabalho é feita pelo professor e pelos próprios elementos do grupo, que realizam uma avaliação dos seus companheiros, assim como uma auto-avaliação.

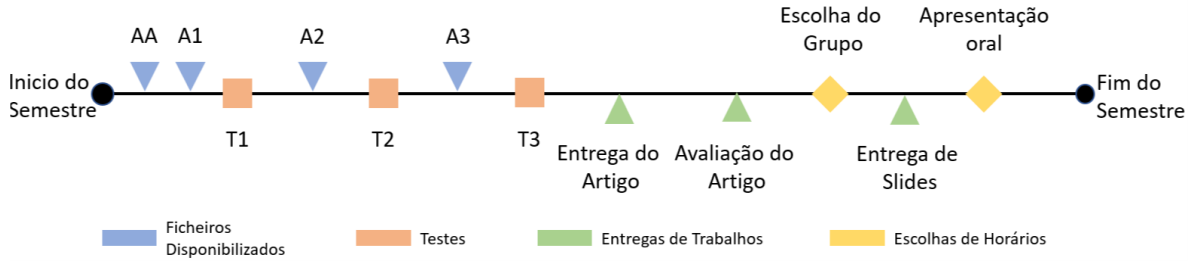


Figura 4.1: *Timeline* dos eventos da unidade curricular.

4.1.2 Atividades Disponibilizadas no Moodle

Usando a plataforma, os estudantes têm acesso a 15 atividades. Todas as observações registadas, pelo Moodle, no *log file* têm um dos seguintes tipos de atividade:

- **Escolha de grupo**, leva a uma página de seleção, onde os estudantes podem criar grupos ou juntar-se a outros estudantes que frequentem a unidade curricular de maneira a formar pequenos grupos de quatro a cinco elementos;
- **Ficheiro**, esta atividade contém vários ficheiros em formato pdf, que por sua vez são constituídos por informação sobre a matéria lecionada durante as aulas teóricas. Esta é subdividida em 4 componentes: Aula de Apresentação, Aula 01, Aula 02 e Aula 03;
- **Fórum**, os estudantes visitam esta atividade para participar em atividades relacionadas com o fórum. Criar novos tópicos de discussão ou participar em tópicos previamente abertos ou visualizar tópicos fechados;
- **Notas finais**, permite o *download* de um ficheiro que contém a classificação obtida na submissão, no trabalho de grupo e nos três testes realizados pelos estudantes;
- **Proposta de tema para os slides e apresentação**, os estudantes discutem vários temas sobre os quais incide o trabalho e utilizam esta atividade para indicar qual é o tema proposto;
- **Registo para apresentação oral**, estando todos os elementos do grupo de acordo, um elemento é responsável por visitar esta atividade e indicar qual será o dia da realização da apresentação oral do grupo;

- **Submissão de slides**, é a atividade onde um elemento do grupo efetua a submissão do conjunto de slides que foi criado pelo grupo;
- **Submissão e revisão de artigos**, os estudantes visitam esta atividade para submeter os seus artigos, antes da data de entrega. Após esta data, os artigos são avaliados pelo professor e por um grupo de estudantes. A seleção do grupo de estudante para avaliação, é realizada aleatoriamente pela plataforma. A nota de avaliação do artigo corresponde à combinação entre a nota do professor e a nota atribuída pelo grupo de estudantes;
- **Template**, o professor torna disponível, na plataforma, um ficheiro em formato pdf que contém algumas diretrizes sobre como deve ser escrito o artigo, que será submetido mais tarde;
- **Teste**, é constituída de três instâncias, Teste 01, Teste 02 e Teste 03. Os estudantes visitam estas atividades no dia do teste como forma de iniciar a sua avaliação;
- **UC info**, contém toda a informação que se encontra relacionada com a unidade curricular. Número de estudantes, previsão de horas de trabalho, informação sobre o professor, objetivos, método de avaliação, *work plan*, conhecimento adquirido e bibliografia.

4.1.3 População e Amostra

4.1.3.1 População

A "população" é constituída por todas as ações realizadas pelos estudantes registados na unidade curricular que utilizaram a plataforma, para aceder a recursos ou participar em atividades no âmbito da unidade curricular de Comunicação Técnica.

4.1.3.2 Amostra

A amostra contém toda informação registada pela plataforma, acerca de todas as atividades realizadas pelos estudantes, durante os anos letivos 2015/16, 2016/17 e 2017/18. No total existem 90416 observações, relativas às atividades de 522 estudantes.

4.2 Preparação de Dados

A ferramenta de recolha de dados usada foi o Moodle. A recolha de dados foi feita automaticamente pela plataforma, cada vez que esta foi usada por um estudante. Todas as ações são registadas pelo sistema, através da criação de uma nova entrada no ficheiro de *log*. Este método garante que o *log file*, contém todas as atividades realizadas na plataforma desde o início até ao fim do semestre, desde 2015 até 2018.

Os *log files* gerados pela plataforma têm o formato .xlsx, Microsoft Excel Open XML Spreadsheet.

4.2.1 Formato Inicial dos Dados

O *dataset* inicial é constituído por nove campos, automaticamente gerados pelo Moodle:

- **Tempo**, representa a data da entrada no formato, dd/mm/yyyy, hh:mm;
- **Nome completo do utilizador**, registo do nome do estudante que realizou a ação;
- **Utilizador afetado**, nome do utilizador sobre o qual a ação incidiu, pode ser preenchido ou deixado em branco, quando aplicável;
- **Contexto do evento**, é armazenado o nome da atividade onde a ação ocorreu (Ficheiro: Aula02);
- **Componente**, regista o tipo de atividade visitada das referidas em 4.1.2 (Ficheiro);
- **Nome do evento**, cada ação tem uma breve descrição sobre a ação que ocorreu, (Módulo de unidade curricular visualizado);
- **Descrição**, contém uma frase escrita em inglês ou português, que faz um retrato do que se passou (The user with id '21104' viewed the 'resource' activity with the course module id '65007');
- **Origem**, descreve o tipo de dispositivo usado para o acesso ao Moodle, "cli" ou "web";
- **Endereço IP**.

Para se realizarem as previsões foi necessário, a criação de novos campos que contêm informações extraídas dos dados originais. Para tal, foi iniciado o processo de preparação e consequente limpeza dos dados, descrito na secção 2.4.

4.2.2 Transformação dos Dados

Os campos iniciais e a informação contida neles foi considerada insuficiente para a criação do modelo. Existiu, assim, a necessidade de criar mais campos que contêm informação extraída diretamente dos dados recolhidos. Foram criados 14 campos, passando o número total de campos a ser 20. Uma vez que os campo "Tempo", "Nome completo de utilizador" e "Descrição", são eliminados e substituídos durante o processo de decomposição.

4.2.2.1 Desconstrução de Tempo

A informação contida no campo Tempo segue o formato, dd/mm/yyyy, hh:mm, é possível fazer a divisão deste campo em três, o que facilita o acesso e a identificação das observações:

- **Dia de Início**, contém o dia e o mês registado, dd/mm;
- **Hora de Início**, inclui a hora e os minutos quando foi visitada a plataforma, hh:mm;
- **Ano**, regista o ano, yyyy.

4.2.2.2 Identificação de Eventos

Assim como o campo Tempo foi dividido em vários componentes, a mesma lógica foi aplicada para os eventos. A frase contida na descrição é demasiado extensa e, como tal, contém bastante informação sobre o estudante (Nr de Estudante), o que ele fez (Ação), que atividade visitou (Atividade, Id da Atividade e Atividade Afetada), e tendo acesso ao Id da Atividade, é possível saber qual o Nome da Atividade.

O texto contido no campo da Descrição, pode ser escrito em inglês ou português. Apenas alguns casos, de cerca de 500 entradas, é que estavam escritas em português. As restantes 89916 observações estavam escritas em inglês. As frases escritas neste campo seguem um padrão de escrita, que foi identificado e usado para fazer o *parse* da informação contida e dividi-la em seis novos sub-campos.

Como exemplo, considere-se a seguinte frase:
`The user with id '23127' viewed the 'page' activity with the course module id '80587'.`

O padrão garante que o Nr de Estudante está entre a *sub-string* "The user with id ' " e o apóstrofe seguinte. Já o Id da Atividade encontra-se sempre no fim da frase entre dois apóstrofes. Sabendo o Id da Atividade, é possível consultar uma tabela que contém os nomes de todas as atividades e os ids atribuídos, para determinar qual é o Nome da Atividade.

Seguindo o exemplo, obtemos os seguintes dados.

- **Nr de Estudante** = 23127;
- **Id da Atividade** = 80587;
- **Nome da Atividade** = Notas Finais.

Como os campos anteriores já se encontram preenchidos, deixa de ser necessário conter toda essa informação na frase da Descrição. Como tal, todas as *sub-strings* que contêm o Nr de Estudante e Id da Atividade, foram eliminadas. Para tal foram removidas as *sub-strings* que se encontram antes do Nr de Estudante e após o segmento, *with the course module id.*

Após eliminada a informação não relevante, a frase exemplo passa a ser:
`viewed the 'page' activity`

O padrão estabelece que a primeira palavra nesta nova frase, constitui a Ação, que foi registada. As palavras que se encontram compreendidas entre o *the* e a palavra *activity*, formam a Atividade. Tudo o que sucede a Atividade até ao final da frase, constitui a Atividade Afetada.

Segundo o exemplo, ficamos com os dados:

- **Ação** = viewed;
- **Atividade** = page;
- **Atividade Afetada** = (no caso analisado não existe).

A figura 4.2 serve para indicar a posição de cada campo na frase usada como exemplo.

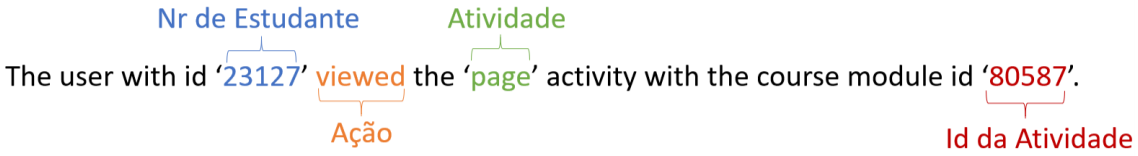


Figura 4.2: Desconstrução do campo da Descrição.

No final deste processo, o campo de Descrição deixa de existir, porque toda a informação contida nele foi dividida e colocada em campos adicionais. Assim, o Nome completo do utilizador, deixa de existir e passa a ser substituído pelo Nr de Estudante, garantindo assim a anonimização.

4.2.3 Flags

Existem três campos que foram usados apenas, para auxiliar os cálculos e a posterior geração de gráficos. São eles o Visto, Manteve na Atividade e Tempo de Sessão. Os dois primeiros campos apresentam valores booleanos, sendo inicializados a False.

O campo Manteve na Atividade apenas se torna True se a Duração de Sessão for menor ou igual a 30 minutos. Este tempo foi considerado suficiente para garantir que o estudante se manteve na sessão, por outras palavras nenhum estudante passou mais de 30 minutos ativamente numa dada página da plataforma.

O campo Visto serve para auxiliar no processo de determinação da Duração da Sessão. Quando uma determinada entrada é consultada, o valor do campo Visto passa a True, indicando que essa observação foi usada no cálculo da Duração da Sessão.

Caso a Duração da Sessão seja menor ou igual a 30 minutos, esse valor passa a ocupar o Tempo de Sessão equivalente. Esta medida serve para garantir que se consegue obter uma observação

que apenas contém a duração real da sessão. Esta informação é bastante pertinente para a criação de um modelo de previsão e de gráficos.

4.2.4 Cálculo da Duração da Sessão

Uma vez que o Moodle não faz o registo de quando é que o estudante abandona a plataforma e, como tal, não é capaz de determinar quanto tempo demorou a sessão do estudante, existiu a necessidade de serem realizados cálculos de forma a determinar a Duração de Sessão e a Hora de Fim. Estes campos encontram-se relacionados uma vez que o último ajuda a determinar o primeiro. Para determinar a Hora de Fim e a Duração da Sessão, foram consultados os *log files* e verificou-se qual o valor da Hora de Início da entrada imediatamente a seguir àquela que se pretende determinar a duração. Uma vez identificada a entrada, sabemos que a Hora de Início da entrada seguinte, será a Hora de Fim da entrada cuja duração queremos determinar. Subtraindo a Hora de Fim à Hora de Início obtemos a Duração da Sessão. Foi também criada uma restrição, em que é garantido que nenhuma Duração de Sessão excedeu seis horas. As análises, feitas, concluíram que nenhum estudante passou mais de seis horas numa atividade, sem ter saído da sessão, ou então visitado outra. O autómato da figura 4.3 e o pseudo código 4.2.4, demonstram o processo de cálculo da Duração de Sessão.

O *dataset* para cada entrada (index$i$) percorre as observações, desde da posição atual (index$j$) até à posição 0 (S1). Quando é encontrada uma entrada com o mesmo Nr de Estudante que o pesquisado (S2), é feita uma verificação para garantir que não tenham passado mais de 6 horas entre as duas entradas (dif). Posteriormente, se a dif não for maior do que 30 minutos (S6) o valor é adicionado ao campo Duração da Sessão da entrada solicitada. A Hora de Fim de uma entrada é calculada somando, a Hora de Início com a dif. Caso o valor da dif seja superior a 30 minutos (S5), a Duração da Sessão passa a ter o valor do tempo médio passado nessa atividade cálculos estes que são feitos durante o processo de análise dos dados. Esse valor é determinado pelo cálculo do tempo médio passado por todos os estudantes nessa atividade. Caso o valor da dif seja superior a 6 horas (S3) então o estudante já não se encontra na mesma sessão ficando a Duração de Sessão e Hora de Fim com o valor original de vazio.

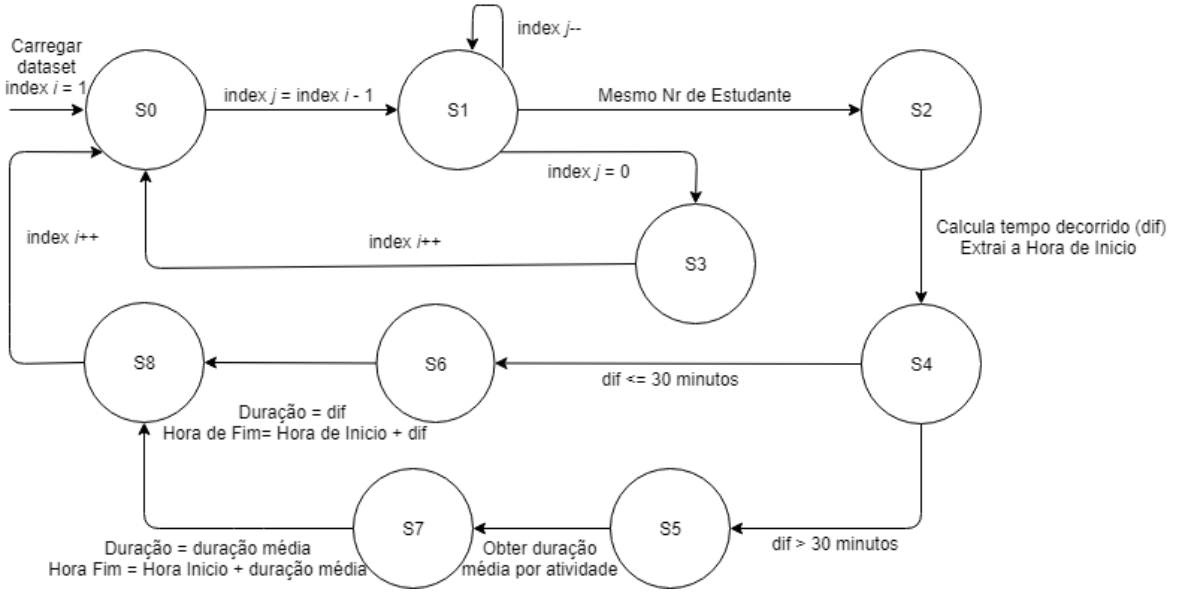


Figura 4.3: Autómato demonstrando o cálculo da Duração da Sessão.

Algorithm 1 Cálculo da Duração da Sessão.

for i = 0 Para todas as observações **do**
  if $entry_i.visto$ = FALSE **then**
    for j = i-1 Até 0 **do**
      if $entry_j.id = entry_i.id$ **then**
        $dif \leftarrow entry_j.h\_inicio - entry_j.h\_inicio$
        if dif <= 6 horas **then**
          if dif <= 30 mins **then**
            $entry_i.duracao \leftarrow dif$
            $entry_i.h\_fim \leftarrow entry_j.h\_inicio$
            $entry_i.manteve\_na\_actividade \leftarrow TRUE$
            $entry_i.tempo\_sessao \leftarrow dif$
          else
            $entry_i.duracao \leftarrow 30\ mins$
            $entry_i.h\_fim \leftarrow entry_i.h\_inicio + 30\ mins$
            $entry_i.visto \leftarrow TRUE$
        else
          break
for i = 0 Para todas as observações **do**
  if $entry_i.duracao$ = EMPTY **then**
    for j = 0 Para todas as observações médias **do**
      if $entry_i.atividade = t\_medio\_atividade_j.atividade$ **then**
        $entry_i.atividade \leftarrow t\_medio\_atividade_j.duracao$
        $entry_i.h\_fim \leftarrow entry_i.h\_inicio + entry_i.duracao$

4.2.5 Limpeza e Organização dos Dados

Como acontece em muitos processos de *data mining*, nem todas as observações são úteis para "treinar" o modelo de Machine Learning, uma vez que muitas vezes contêm erros, valores ausentes, etc. Neste caso foram eliminadas várias entradas, como: todas as ações efetuadas pelo professor ou assistentes do curso. Também foram removidas todas as entradas de estudantes que desistiram da unidade curricular antes da sua conclusão. Este tipo de dados foi removido porque não iria contribuir com nenhuma informação relevante para a geração do modelo, criando até ruido.

A última fase é a de organização dos dados, em que todas as observações foram ordenadas de forma a garantir que todas as entradas, realizadas pelo mesmo estudante, se encontram agrupadas sequencialmente. Este processo é útil, uma vez que permite a deteção de anomalias nas interações. A ordem de organização agrega todas as entradas, primeiro por nome do estudante, depois por ano, por data no sentido ascendente, ex: o primeiro registo de um dado estudante, corresponde sempre à primeira entrada que este realizou. Todas as restantes entradas correspondem ao resto das atividades durante o ano escolar, onde a primeira entrada equivale à observação mais antiga e a última à mais recente.

4.3 Modelo Preditivo

4.3.1 *Features* Usadas na Previsão

Features ou variáveis independentes, também conhecidas como *predictors* representam critérios que podem ser aplicados a todos os estudantes no *dataset*, por exemplo o número de minutos passados num teste. Estas variáveis são depois usadas pelo algoritmo de Machine Learning para facilitar a classificação/avaliação do percurso académico de um estudante. As variáveis foram criadas tendo por base as heurísticas sobre o comportamento dos estudantes, de boas práticas e como resultado de estudos estatísticos realizados sobres as observações.

Foram criados 20 *predictors* para serem usados neste estudo, que são listados a seguir.

4.3.1.1 *After Time* (AT)

A variável "*After Time*" conta o número de minutos necessários para aceder a uma atividade, depois desta se tornar disponível. No caso da unidade curricular DPI1001, é dividida em AT1 e AT2. Enquanto que AT1 foca-se na atividade de "Escolha de Grupo", AT2 foca-se na atividade de "Registo para a apresentação oral".

4.3.1.2 *After being made Available* (AV)

A *feature* "*After being made Available*" conta o número de dias que o estudante demorou, até aceder pela primeira vez aos ficheiros fornecidos, após estes serem disponibilizados. Esta *feature* é dividida em AV1, AV2 e AV3 cada uma relativa aos três ficheiros disponibilizados "Aula 01", "Aula 02" e "Aula 03".

4.3.1.3 *Before Test* (BT)

Esta *feature* é subdividida em BT1, BT2 e BT3. Cada uma dessas instâncias indica o número de dias, entre o primeiro acesso às atividades "Aula 01", "Aula 02", "Aula 03" e o dia do teste correspondente, "Teste 01", "Teste 02" e "Teste 03".

4.3.1.4 *Clicks*, *Downloads* e *Fórum*

Clicks regista o número total de ações realizadas pelo estudante na plataforma. Download regista o número de *downloads* de ficheiros disponibilizados pelo professor. *Fórum* conta o número de entradas relacionadas com a atividade do Fórum.

4.3.1.5 *Test Time* (TT)

Test Time indica o número de minutos que o estudante demorou a realizar o teste. No caso da DPI1001, existem três instâncias de testes, TT1, TT2 e TT3.

4.3.1.6 *Time In/Out of Danger Zone* (TIDZ/TODZ)

TIDZ e TODZ, representam o número de vezes que uma submissão ocorreu dentro ou fora da *danger zone*, definida como sendo os últimos 10 minutos que antecedem o fim de um prazo de entrega. Como existem dois tipos de submissões na unidade curricular DPI1001, também existem duas instâncias de TIDZ e outras duas de TODZ, que se encontram relacionadas com as submissões dos artigos e do conjunto de slides, feitas pelos estudantes.

4.3.1.7 *Total Time Online* (TTO)

Representa o total de minutos que um estudante passou ativamente na plataforma.

4.3.1.8 Semelhança (SIM)

A *feature* SIM, é um caso especial. Encontra-se relacionada com o processo de compararação de diferentes padrões de interações online e é explicado em maior detalhe na secção 4.3.3.

4.3.2 Análise de Correlação

Comparando cada uma das variáveis torna possível a criação de uma matriz de correlação. Esta matriz mostra quais os coeficientes de correlação entre as *features*. Cada célula desta matriz demonstra qual o grau de correlação entre duas variáveis, figura 4.4.

Se qualquer uma destas células apresentar um valor superior a 30% em valor absoluto, podemos assumir que as variáveis estão (bastante) relacionadas uma com a outra. Nestas situações, é aconselhável remover uma das variáveis.

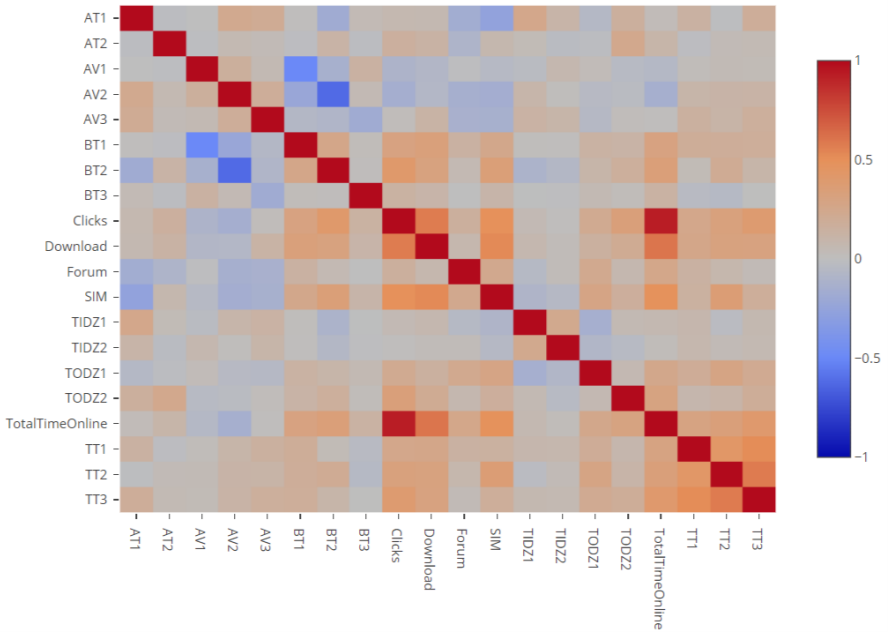


Figura 4.4: Matriz de correlação.

Column 1	Column 2	Correlation ▾
Clicks	TotalTimeOnline	0.92834753095
Download	TotalTimeOnline	0.60336565295
TT2	TT3	0.58086716291
Clicks	Download	0.5769107709
Download	SIM	0.52659479762
TT1	TT3	0.51466288266
Clicks	SIM	0.49732822882
SIM	TotalTimeOnline	0.48790685111
TT1	TT2	0.43713422913

Column 1	Column 2	Correlation ▴
AV2	BT2	-0.6223418255
AV1	BT1	-0.49604581114
AT1	SIM	-0.26008410487
AV2	BT1	-0.22262511199
AV3	BT3	-0.17837245848
AT1	BT2	-0.175944147
AT1	Forum	-0.16410345277
AV2	SIM	-0.15675396885
AV2	Clicks	-0.15171929193

Figura 4.5: Correlações positivas e negativas.

Depois de analisar a matriz, figura 4.4 juntamente com a figura 4.5, um total de cinco *features*, AV1, AV2, TTO, Clicks e Downloads, foram consideradas demasiado similares quando comparadas com as restantes. Todas estas variáveis apresentavam uma correlação superior a 30% em valor absoluto e, como tal, foram excluídas.

As resultantes 15 *features* (listadas na tabela 4.1), foram consideradas válidas e fornecem ao modelo informação, de forma a determinar se o estudante irá reprovar ou não à unidade curricular, tendo como base as suas ações no Moodle.

Tabela 4.1: *Features* que foram consideradas válidas.

Features	Número de instâncias
AT	2
AV	1
BT	3
Forum	1
TIDZ	2
TODZ	2
TT	3
SIM	1

4.3.3 Comparação do Percurso Académico

4.3.3.1 *Feature* Semelhança (SIM)

Enquanto que as *features* descritas anteriormente consistem em informação que é diretamente extraída do *dataset* final, a SIM (*similarity*) é diferente, uma vez que tem de ser extraída indiretamente, usando cálculos.

O objetivo é determinar quão parecido o percurso académico do estudante x é, quando comparado com o percurso mais semelhante do estudante y, cuja nota é superior a 16 valores (numa escala de 0 a 20). Todos os estudantes que obtiveram classificações finais superiores a 16 estão reunidos num grupo denominado *HG*.

A SIM, como uma variável independente, representa a similaridade entre o percurso académico de dois estudantes. Foi criada seguindo a suposição de que se a sequência de atividades realizada por um estudante com uma classificação final elevada, for comparada com a sequência de um outro estudante que siga um padrão de atividades realizadas semelhante, então pode-se concluir que o segundo estudante poderá também obter uma classificação final elevada, semelhante à classificação do primeiro estudante.

Para obter o fator de similaridade, inicialmente ocorreram dois tipos de transformações: primeiro, todas as entradas recolhidas foram transformadas numa string, usando *Run-Length Encoding* (RLE); na segunda fase, essas *strings* foram convertidas para o formato de um ponto representado num espaço geométrico n-dimensional.

4.3.3.2 Transformação usando RLE

O processo de transformação das atividades visitadas numa *string* utilizando RLE é aplicado a todos os estudantes e levou à criação de 103 *strings* por estudante, uma vez que cada semestre é constituído por 103 dias.

Ao fazer a análise dos ficheiros *log*, foi possível identificar todos os tipos de atividades existentes, o tempo médio passado por atividade, o número de visitas e analisar a estrutura/observações da unidade curricular. Com a ajuda do docente, foi possível descobrir as atividades que têm o maior impacto na nota. Como tal, surgiram 10 tipos de atividades centrais que são identificadas como as mais importantes na previsão da nota final:

- Os ficheiros a que os alunos têm acesso (Aula de apresentação, Aula01, Aula 02, Aula 03);
- Os testes (Teste 01, Teste 02, Teste 03);
- Escolha de grupo;
- Submissão e revisão de artigos;
- Notas finais.

De maneira a facilitar a identificação dos diferentes tipos de atividade, após a transformação usando RLE, todos os tipos de atividades passaram a ser identificados por letras do alfabeto, A, B, ... até J. Cada letra é precedida pelo número total de minutos, passados pelo estudante naquela atividade ao longo desse dia.

Cada semestre, na instituição onde foram recolhidos os dados, é constituído por 103 dias. Portanto, de maneira a realizar uma comparação completa e extensiva, os dias considerados, por cada estudante, abrangeram o Dia 1 até o Dia 103. Este período de tempo (*timeframe*) foi considerado preferencial, quando comparado com um *timeframe* que apenas considera os dias em que o estudante executou uma ação, porque garante que todos os estudantes têm o mesmo número de dias no ficheiro *log* e, como tal, permite a realização de comparações mais precisas entre os estudantes.

Após ser feita a codificação, surgiram três tipos distintos de entradas:

- **Tipo *Done*** descreve quanto tempo e quais as atividades visitadas pelo estudante naquele dia. As atividades incluídas neste tipo têm de pertencer àquelas que já foram referidas anteriormente. O *output* obtido é constituído por um número, seguido de uma letra. Esta sequência é repetida n vezes, onde n ϵ {1,2,3,...,10};
- **Tipo *Empty*** é o tipo mais comum, é usado quando o estudante não utilizou a plataforma naquele dia, logo não existe nenhuma observação para esse dia. É representado por uma *string* vazia;

- **Tipo *Not Done*** este tipo ocorre em situações onde os estudantes visitaram uma ou mais atividades naquele dia, mas essas atividades têm tipos diferentes daqueles referidos anteriormente.

4.3.3.3 Calcular a Distância de Interação

A métrica usada para medir a distância de interação é baseada no conceito de distância num espaço ortogonal multi-dimensional. A *string* associada a todos os dias é convertida, usando o formato de um ponto, num espaço de 10-dimensões.

A transformação é feita atribuindo-se uma coordenada a cada letra da *string* e substituindo-se a posição da referida coordenada pela quantidade de tempo passado nessa atividade, pelo estudante.

Exemplo 1

`Considerando que existe a` *`string`* `11A3C4I`

`A transformação resulta no ponto (11,0,3,0,0,0,0,0,4,0)`

Após ser feita a transformação de todas as *strings* num ponto pertencente a um espaço de 10-dimensões, é possível estabelecer comparações entre pontos que existem nesse espaço, usando a distância Euclidiana.

Calcular a distância entre dois pontos, serve para determinar a proximidade nos padrões de interação de dois estudantes: quanto menor a distância, mais próximos os pontos estão uns dos outros; por sua vez, isso indica se uma determinada sequência é ou não semelhante a outra.

Depois de aplicado este processo ao estudante x, é possível fazer a comparação entre o estudante x e todos os estudantes $y \epsilon HG$. Terminada a comparação, podemos identificar qual o estudante y cujo percurso de atividades do Moodle é o mais semelhante ao do estudante x. Tendo identificado o melhor estudante y, com o qual fazer a comparação, é altura de realizar a comparação em si. Aplicando a fórmula matemática 4.1, é possível calcular a variação que ocorreu num único dia de atividade do estudante x, quando comparado com o seu par ideal no conjunto HG.

$$SIM_{x\ddot{y}} = \sum_{i=1}^{103} \left(\alpha_i + \frac{|\ \Delta t_i\ |}{3600} \right), \forall x \in estudantes \wedge \exists \ddot{y} \in HG : min|x - y| = d(x, \ddot{y}) \tag{4.1}$$

As comparações são calculadas usando o número de diferentes atividades visitadas pelo estudante (α) e a quantidade de tempo passado nessa atividade, quando comparado com o tempo passado pelo par ideal (Δt), fórmula 4.2. Este valor é depois dividido pelo número de segundos numa hora.

$$\Delta t_i = \sum_{j=1}^{10} \left(|t_{jx} - t_{j\ddot{y}}| \right), \forall x \in estudante \wedge \ddot{y} \in HG \tag{4.2}$$

Exemplo 2

```
Dadas duas sequências:

Estudante X = 12A56B6J

Estudante Y = 11A64B7J
```

Como podemos ver, em ambos os casos foram visitadas as mesmas atividades, logo $\alpha = 0$. O valor $\Delta t = ((|12 - 11|) + (|56 - 64|) + (|6 - 7|))$

Aplicando a fórmula 4.1 obtêm-se um valor de $\approx .00278$.

Exemplo 3

```
Dadas duas sequências:

Estudante X = 6A38C13G

Estudante Y = 64C
```

Neste exemplo as sequências são diferentes. A sequência do estudante Y é muito menor do que a do estudante X. Uma vez que existem duas atividades diferentes, o valor de $\alpha = 2$, porque ambas as atividades A e G estão ausentes na segunda sequência. Δt é calculado subtraindo as durações das atividades iguais e somando o tempo das sequências que não estão presentes, $(6 + (|38 - 64|) + 13)$.

Aplicando a fórmula 4.1 obtêm-se um valor ≈ 2.0125.

Se o valor $SIM_{x\ddot{y}}$ obtido for baixo indica uma variação pequena em relação ao caminho académico do estudante com uma nota elevada. Logo, podemos assumir que o estudante x tem uma grande probabilidade de atingir uma boa classificação.

A culminação de similaridades, resulta da soma de todos os valores obtidos em cada dia, sendo que o valor resultante corresponde à nova *feature* SIM.

Quanto mais pequeno for o valor, mais parecida é a sequência de atividades de um estudante quando comparado com o estudante pertencente ao grupo *HG*, com o qual a comparação foi feita. É possível concluir que quanto menor for o valor do SIM, maior é a probabilidade de um estudante obter uma nota elevada.

4.4 Resumo

Neste capítulo foram: explicadas a estrutura e organização da unidade curricular; além disso, foram descritos quais os dados iniciais presentes em cada amostra; foi também justificada e explicada a criação de novos campos; e por fim, foi explicado todo o processo de criação de variáveis independentes, que foram usadas para realizar as previsões.

No próximo capítulo é descrito o processo de criação da variável objetivo e da árvore de decisão, juntamente com os resultados obtidos após serem realizados os testes e quais são as limitações do modelo.

Capítulo 5

Resultados e análise

5.1 Variável Objetivo

Um dos objetivos desta dissertação foi definir uma metodologia que crie um modelo capaz de prever a nota de um estudante, durante o semestre, usando apenas a informação sobre as interações com o *Learning Management System* (LMS). A integração do modelo com um sistema de alerta, representa algo benéfico, tanto para o estudante como para o professor. Esta ferramenta proporciona a possibilidade de, no decorrer do semestre, ser emitido um alerta quando o padrão de interações de um estudante pode conduzir a uma reprovação.

Uma vez que o modelo se encontra focado em prever se um estudante vai ou não reprovar à unidade curricular, a variável objetivo ou "alvo", não precisa de incluir todos os valores na escala de avaliação da instituição (0 - 20). É de relembrar que a metodologia descrita, serve para emitir alertas ou avisos. Em vez disso, para atingir os objetivos propostos, apenas é necessário definir duas categorias de notas: as de aprovação e de reprovação.

O modelo distingue-se de outros porque não considera classificações obtidas durante avaliações intermédias (testes), focando-se apenas nos padrões de interação entre o estudante e a plataforma. Esta característica é benéfica, considerando que nem todas as disciplinas utilizam o Moodle para fazer as avaliações ou a componente da avaliação intermédia pode nem existir no contexto da disciplina e as avaliações podem ser só feitas por exame final, sem recurso a testes.

Como tal, foram criadas três categorias para a nota resultante, descritas na tabela 5.1. Usando este tipo de agrupamento, a categoria A foi definida como sendo uma potencial nota de reprovação ou negativa, a categoria B representa as notas instáveis ou incertas uma vez que o estudante pode muito facilmente alternar entre uma nota de reprovação/aprovação, consoante variações muito pequenas, e a categoria C classifica um bom comportamento nas interações entre o estudante e o sistema.

Tabela 5.1: Categorias alvo.

Categoria	Intervalo de nota (na escala de 0 a 20)
A	0 - 8
B	9 - 11
C	12 - 20

5.2 Árvore de Decisão Gerada

Após definir qual é a variável objetivo, um algoritmo do tipo árvore de decisão pode ser aplicado ao *dataset*, usando as três categorias como sendo a variável objetivo e as variáveis independentes como sendo preditores.

Para aplicar à árvore de decisão, o algoritmo tem primeiro de ser treinado. Para tal, o *dataset* é dividido em dois subconjuntos diferentes, o conjunto de dados de treino, que contém 70% do conjunto de dados original e os restantes 30% constituem o conjunto de dados de teste. Para garantir que todas as categorias se encontram igualmente representadas de acordo com o balanceamento de categorias, no conjunto de teste foi usado *over* e *undersampling*. Criou-se assim um *dataset* que contém 327 entradas, onde cada categoria é representada, individualmente, por 109 observações.

O conjunto de dados de treino serve para ensinar ao modelo como identificar padrões de interação e qual a classificação que deve ser atribuída. Inicialmente decorre a fase de "treino" do algoritmo, onde este analisa o *dataset* de treino e usa os dados fornecidos para "aprender" a determinar a variável objetivo. Posteriormente o algoritmo analisa as variáveis independentes e determina qual é a categoria resultante.

A fase seguinte envolve o teste do modelo. Depois do algoritmo ser ensinado, é usado o conjunto de teste, de forma a determinar qual a sua qualidade preditiva. O algoritmo analisa as *features*, identifica padrões de interação e realiza previsões sobre qual será o valor da variável objetivo, usando o conhecimento prévio como base para as previsões. Terminado este processo, é feita uma comparação entre os valores da variável objetivo obtidos e os valores reais. A comparação gera os dados estatísticos da tabela 5.2. No final de todo este processo, é criada uma imagem, figura 5.1, que representa a árvore de decisão gerada. Analisada a imagem, conseguimos entender as decisões tomadas pelo algoritmo.

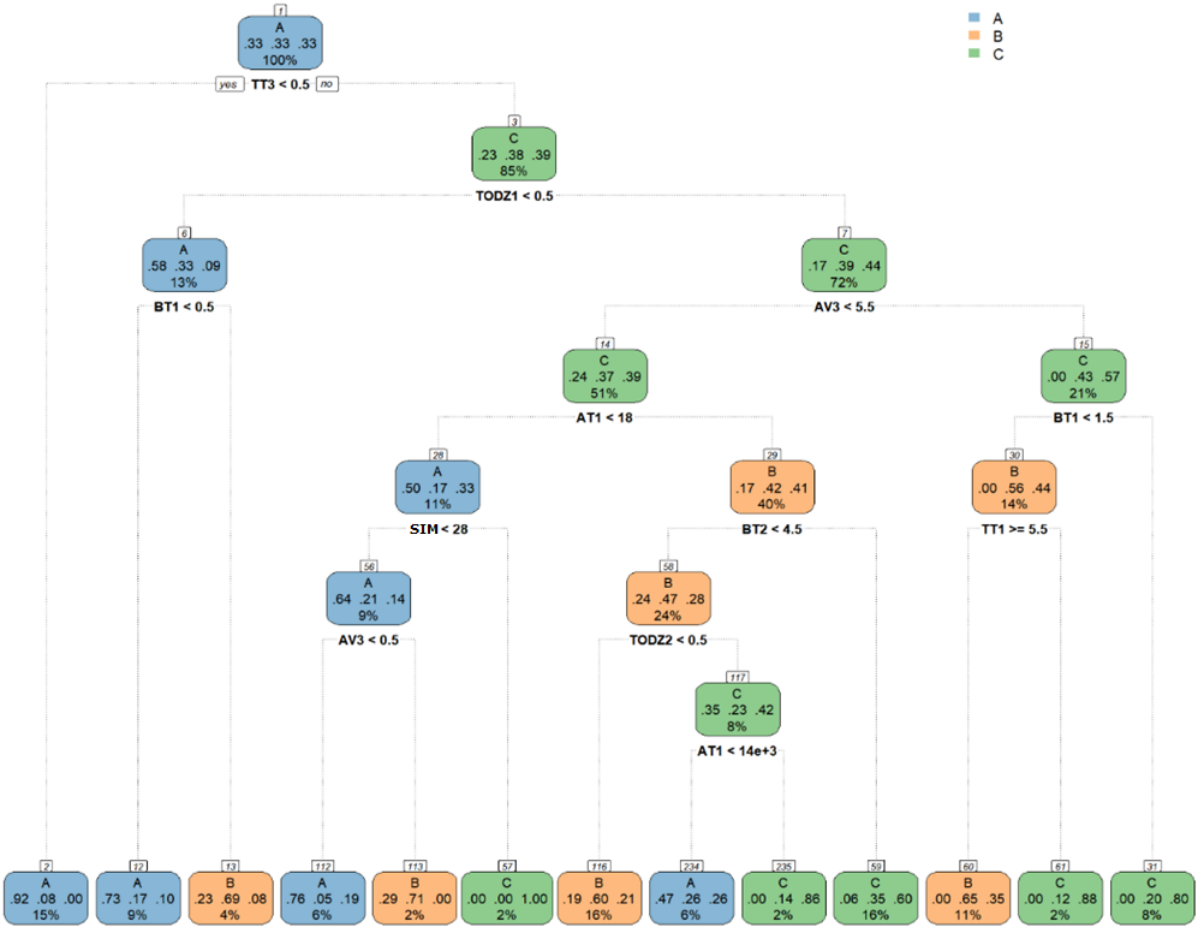


Figura 5.1: Árvore de decisão obtida.

Em cada nó da árvore, encontra-se uma condição, baseada nas variáveis independentes. Todos os nós/folhas que se seguem respeitam ou não o requisito anterior. Portanto, as folhas ou nós que se encontram à esquerda de uma condição respeitam a mesma condição. Em termos booleanos obtêm valor verdadeiro. As folhas que estão à direita não respeitam essa condição e, como tal obtêm valor falso.

Analisando a árvore de decisão criada (figura 5.1), é possível reparar que as três categorias da variável objetivo se encontram representadas nos nós terminais/folha da árvore, o que indica que o modelo consegue identificar os três tipos de categorias. Estudando a árvore, é possível entender o processo de previsão e quais as decisões (*features*) tomadas em cada, nó de maneira a classificar as interações do estudante e determinar uma classificação. Além disso é possível determinar, qual a importância relativa de cada variável independente durante o decorrer do processo de decisão.

Podemos ver que o algoritmo consegue distinguir bem quando é que foi obtida uma classificação do tipo A. Logo nos nós iniciais, e com pouca profundidade, ele é capaz de determinar se um estudante tem uma nota negativa. Apesar da maior abundância de observações da categoria C, em relação às restantes, em específico à categoria B, a distinção entre as duas ocorre, como

podemos ver, a profundidades maiores que três.

5.3 Importância das *Features*

Nem todas as *features* criadas foram usadas na árvore de decisão. Apenas nove foram consideradas importantes o suficiente para serem usadas na árvore de decisão. A figura 5.2, demonstra todo o percurso das *features*, desde a sua criação ao seu uso.

Podemos ver que inicialmente foram criadas 20 variáveis independentes. Após a validação da correlação, foram eliminadas cinco *features*, estas foram consideradas muito relacionadas às restantes e por isso foram eliminadas. Uma vez aplicadas as 15 variáveis independentes, a árvore de decisão escolheu usar nove no processo de previsão, estas nove finais foram consideradas as mais relevantes.

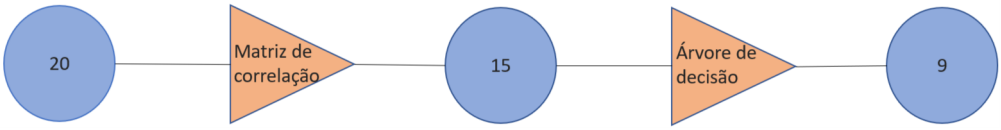


Figura 5.2: Seleção das *features* a serem usadas.

O gráfico da figura 5.3 mostra qual a importância de cada variável independente, no processo de previsão da árvore de decisão. Como podemos ver a variável TT3 é a que apresenta maior importância, a justificação para esse valor relaciona-se com o facto de que estudantes com notas negativas, em média passam menos tempo durante a duração do último teste. Isto deve-se provavelmente à má preparação dos estudantes ou então porque eles próprios já sabem que, devido aos seus comportamentos ao longo do semestre não há muito que possam fazer para atingir uma nota positiva. Como tal, o tempo que passam no terceiro teste é muito menor, quando comparado como os restantes estudantes.

Ambos os TODZs e TIDZs, são excelentes indicadores do carácter do estudante. Pode-se assumir que, apenas os estudantes que não estão preparados ou cientes dos prazos de entrega, fazem submissões com menos de 10 minutos até ao final da entrega. Este tipo de ação demonstra um comportamento de risco, que ajuda o sistema a distinguir entre estudantes de categoria A dos da categoria C e categoria B da C. No geral, pode-se assumir que estudantes que realizam entregas dentro da "danger zone", são mais propensos a ter uma classificação negativa.

A *feature* SIM ajuda a fazer a diferenciação entre um estudante que tem uma nota negativa ou positiva. Essa importância é justificada com as diferenças nos comportamentos das duas categorias de estudantes, uma vez que o comportamento de um estudante que teve negativa varia bastante do comportamento de um estudante que teve uma nota positiva.

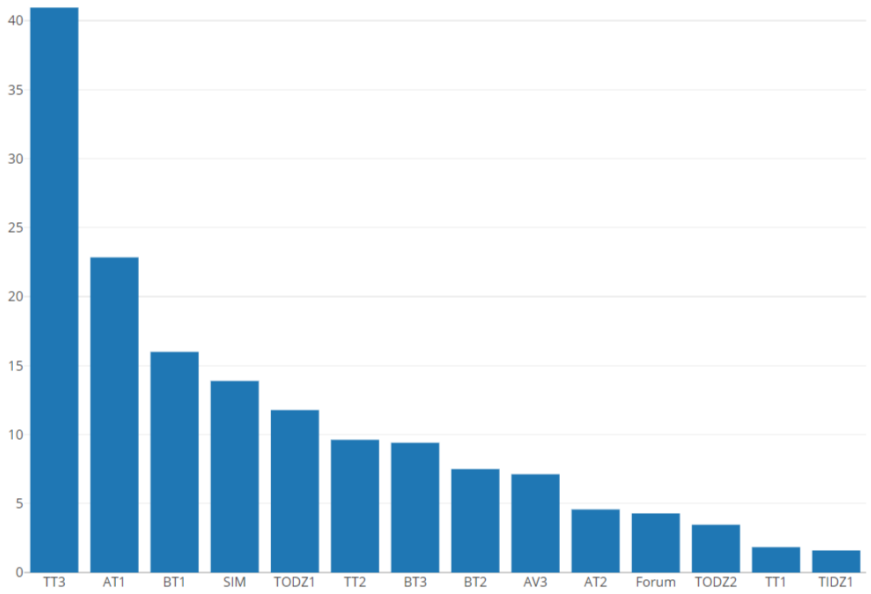


Figura 5.3: Importância de cada *feature* na árvore de decisão.

5.4 Resultados Obtidos

Para a obtenção de resultados foi usado a versão 4.1-13 do R e a versão 5.0.3 do Exploratory[1] que, permite a análise e exploração dos dados numa única plataforma.

A tabela 5.2 mostra a relevância e qualidade preditiva do modelo.

Tabela 5.2: Avaliação da qualidade preditiva do modelo com 100% dos dados.

Semestre	Categoria	*F-Score*	*Accuracy*	Taxa de Erros	*Precision*	*Recall*
100%	A	**82**%	**86**%	14%	72%	**96**%
	B	59%	77%	23%	74%	49%
	C	67%	78%	22%	67%	67%

Analisando a tabela 5.2, podemos concluir que o modelo, no geral, é bastante eficiente a prever todos os tipos de categorias, uma vez que apresenta uma *accuracy* média de 80,33%. O modelo demonstra uma elevada *precision* e baixa *taxa de erros*. Estes valores demonstram também a capacidade do modelo para identificar corretamente todas as categorias.

Para a categoria A os valores altos de *accuracy* e *precision* mostram que o modelo é capaz de prever corretamente qual será o tipo de categoria a atribuir aos estudantes. Um valor elevado de *F-Score* mostra que o modelo é eficaz quando prevê a categoria A. A elevada taxa de *recall*, demonstra que o modelo deteta um número elevado de casos das categorias em causa, enquanto que uma baixa taxa de erros garante que o modelo é fiável.

[1]Link: https://exploratory.io/

Contudo, é relevante indicar que o modelo tem alguns problemas de distinção quando tenta classificar o comportamento dos estudantes como conduzindo a uma nota do tipo B ou C, apesar de ambos os casos apresentarem valores elevados de *precision* e *accuracy*. Esta dificuldade pode ser justificada com baixos valores obtidos pelo *recall* e *F-Score* dessas duas categorias. Para além disso, a similaridade nos comportamentos de ambos os grupos pode estar na origem da confusão.

5.5 Robustez à Falta de Informação

A última fase deste projeto foi testar se o modelo pode ser usado para prever as notas dos estudantes com exatidão, durante o decorrer do semestre. Uma vez demonstrado que, no final do semestre, o modelo é capaz de obter previsões precisas, foi decidido realizar um conjunto de testes onde o número de dias, disponíveis nos *logs*, é reduzido, de forma a determinar qual seria a resposta do modelo caso o número de observações fosse menor.

O *dataset* original foi copiado três vezes e o número de dias presente em cada cópia foi reduzido em -25%, -50% e -75%, respetivamente. Como cada semestre é constituído por 103 dias, quer dizer que os novos *datasets* tinham observações relativas aos primeiros 77, 51 e 26 dias. Estes três *datasets* usados para o treino do algoritmo, foram novamente divididos em dois subconjuntos: o de teste e de treino, usando-se *under* e *oversampling* conforme a necessidade. A figura 5.4 mostra qual a *accuracy* obtida para cada categoria ao longo do semestre, assumindo a divisão do semestre em 4 frações.

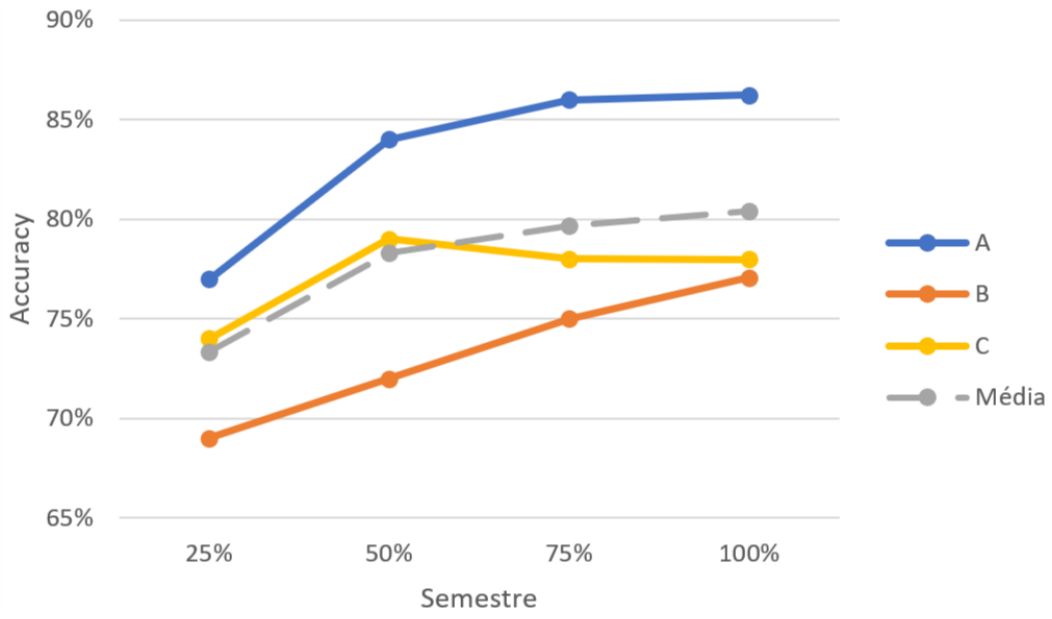


Figura 5.4: Evolução da *accuracy* do modelo ao longo do semestre.

Ao analisar a figura 5.4, podemos concluir que, apesar da redução no número de dias a que o modelo tem acesso este ainda é capaz de atingir o seu objetivo principal, de prever com exatidão quais os estudantes que irão ter notas de reprovação (categoria A).

Pode-se também verificar que a *accuracy*, para as categorias A e B, tem tendência de aumentar ao longo do semestre, sendo apenas a categoria C que sofre uma pequena descida de cerca de 1%, quando se encontra na segunda metade do semestre. De notar que para as classificações negativas, o algoritmo consegue prever com uma *accuracy* mínima de 77% e com um valor máximo de 86%, demonstrando a capacidade do modelo de obter previsões de classificações negativas corretamente. O valor médio da *accuracy* cresce logaritimicamente ao longo do semestre, começando nos 73% e terminando nos 83%.

Contudo, para as classificações do tipo C, o modelo atinge o seu valor máximo de 79% a meio do semestre e este valor decresce até à sua conclusão. Já o valor da *accuracy* para o tipo B, cresce ao longo do tempo. Os valores semelhantes obtidos pelo modelo no final do semestre, relativamente às categorias do tipo B e C, demonstram a dificuldade que o modelo tem em distinguir certos comportamentos como sendo do tipo B ou C. Esta conclusão é apoiada pelos restantes dados e os resultados obtidos, demonstrando assim uma das fragilidades do modelo.

Usando o valor da média e os restantes resultados, pode-se classificar o modelo, como sendo capaz de prever classificações ao longo do semestre e capaz de identificar quando é que o estudante vai obter potencialmente uma classificação negativa.

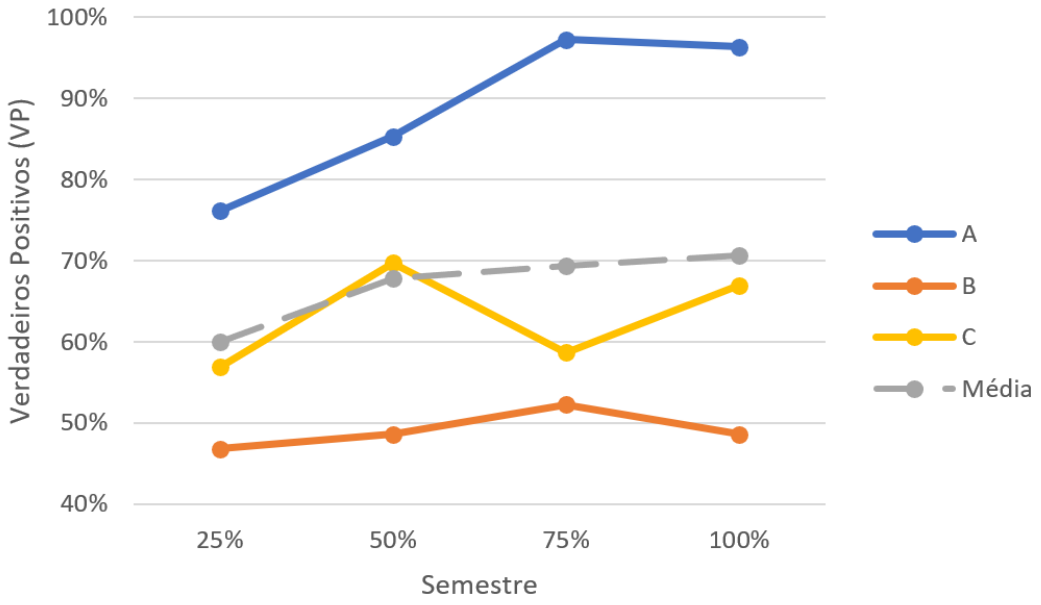


Figura 5.5: Número de verdadeiros positivos obtidos ao longo do semestre.

Na figura 5.5, tanto a média como o número de Verdadeiros Positivos (VP) da categoria A, crescem à medida que o semestre decorre. Contudo, o crescimento do número de VP da categoria C sofre uma pequena redução de aproximadamente 10%, quando são usados -25% dias. Ao mesmo tempo, os valores da categoria B aumentam 3%. Este comportamento pode ser justificado

com a similaridade dos padrões de interação de ambos os estudantes à medida que o semestre chega ao fim. Chegando ao fim do semestre, (100%), os resultados normalizam-se, a categoria A mantém os valores elevados, enquanto que o número de VP da categoria B diminui e o da categoria C aumenta.

Podemos concluir que estes testes servem para provar que o modelo é bastante eficaz a prever notas do tipo A, com um valor base de 76%, que aumenta à medida que o número de dias também aumenta, sendo o valor médio de 89%. Portanto, este modelo pode ser implementado num sistema de alertas para ambos os estudantes e professor, de forma a avisá-los sobre os comportamentos que podem conduzir a uma nota negativa, dando a ambos a oportunidade de alterar os comportamentos do estudante, de maneira a este obter a aprovação à disciplina

Tabela 5.3: Avaliação da qualidade preditiva do modelo com 25% dos dados.

Semestre	Categoria	*F-Score*	*Accuracy*	Taxa de Erros	*Precision*	*Recall*
25%	A	69%	77%	23%	63%	76%
	B	50%	69%	69%	31%	47%
	C	59%	74%	74%	26%	57%

Tabela 5.4: Avaliação da qualidade preditiva do modelo com 50% dos dados.

Semestre	Categoria	*F-Score*	*Accuracy*	Taxa de Erros	*Precision*	*Recall*
50%	A	78%	**84**%	16%	73%	**85**%
	B	54%	72%	28%	61%	49%
	C	69%	79%	21%	68%	70%

Tabela 5.5: Avaliação da qualidade preditiva do modelo com 75% dos dados.

Semestre	Categoria	*F-Score*	*Accuracy*	Taxa de Erros	*Precision*	*Recall*
75%	A	**82**%	**86**%	14%	71%	**97**%
	B	58%	75%	25%	66%	52%
	C	64%	78%	22%	71%	59%

As tabelas 5.3, 5.4 e 5.5, demonstram, a evolução dos, resultados obtidos à medida que aumenta o número de dados fornecidos ao longo do semestre. Ao analisar estes resultados, pode-se ver que os valores obtidos de *F-Score*, *accuracy*, *precision* e *recall*, aumentam à medida que o semestre decorre, já o valor da Taxa de Erros tem tendência a diminuir. Estes resultados mostram que com o decorrer do semestre, à medida que aumenta do número de dados disponibilizados, o modelo torna-se mais eficaz a realizar previsões. Esta conclusão pode ser também ser derivada da análise das figuras 5.4 e 5.5.

5.6 Limitações do Modelo

A metodologia que originou o modelo leva a que este apresente algumas limitações. A principal limitação está relacionada com o facto desta abordagem ser apenas usada em disciplinas que integram e usam bastante o Moodle, durante todo processo educacional. A disciplina usada como caso de estudo, tinha vários eventos (testes, submissões, fóruns e escolha de atividades) associados com o Moodle o que leva a que os estudantes tivessem obrigatoriamente de interagir com a plataforma, o que por sua vez aumenta o número de observações disponíveis. Isto torna o modelo mais eficaz a identificar padrões de atividade e consequentemente, a prever classificações.

Este modelo pode ser aplicado a outras disciplinas que atribuem a mesma importância ao uso do Moodle (como a disciplina estudada), exigindo algumas alterações mínimas relacionadas com os identificadores internos que são atribuídos às atividades pelo Moodle, uma vez que cada disciplina tem números de identificação únicos.

Contudo, caso o modelo seja aplicado a uma disciplina que praticamente não faz uso do Moodle ou das suas características, usando-o apenas como repositório de material relacionado com a disciplina e como fonte de entregas de trabalhos, então as capacidades preditivas do modelo serão prejudicadas.

Como muitas das variáveis independentes descritas na secção 4.3 se encontram relacionadas com os eventos de testes, fóruns e entregas de trabalhos, podemos concluir que, se uma disciplina não fizer uso destes recursos, significa que o modelo não poderá fazer uso destas *features*, uma vez que elas não registaram nenhuma alteração e, como tal, seria menos preciso e exato nas suas previsões.

Em conclusão, este modelo apenas pode ser aplicado a um grupo específico de disciplinas que utiliza o Moodle extensivamente.

Capítulo 6

Conclusões

Neste documento é descrita uma metodologia que pode ser aplicada a uma *framework*/sistema capaz de emitir alertas para os estudantes e os professores. Estes alertas informam sobre o risco de reprovação de um estudante. Para tal, foi feita a análise dos comportamentos e interações online, dos estudantes, com o *Learning Management System* (LMS) para prever se o estudante vai potencialmente reprovar à disciplina, em qualquer ponto durante o semestre.

O modelo usa *features* criadas com base em dados recolhidos de experiências passadas dos estudantes e utiliza um algoritmo de Machine Learning, que foi treinado com dados relativos a três anos de interações entre os estudantes e o Moodle.

Aplicando as variáveis independentes ao algoritmo de árvore de decisão, é possivel classificar qual a importância de cada variável usada no processo de previsão. Os resultados obtidos durante a avaliação da árvore de decisão claramente mostram que o sistema é, de facto, eficiente a identificar estudantes que se encontram em risco de ser reprovados à disciplina.

Ao longo desta dissertação foram descritos os métodos usados nas transformações aplicadas aos dados originais, de maneira a extrair informação mais precisa sobre as ações do estudante na plataforma e como foram calculados alguns dados indiretos, como a duração das atividades. No final deste processo foi criada uma matriz de 15 *features*, consideradas independentes e válidas que, juntamente com 327 observações, foram usadas para criar uma árvore de decisão capaz de prever as notas finais em três categorias. Estas categorias foram criadas para realçar:

- Situações que potencialmente irão conduzir a uma reprovação na disciplina e, como tal, necessitam da atenção especial do professor e do estudante;
- Notas "transitórias", que podem ser positivas ou negativas;
- As restantes notas que potencialmente são positivas.

Os resultados obtidos mostram que o modelo é capaz de identificar três tipos de situações com uma *accuracy* média de 80%, quando as previsões são feitas no final do semestre e 67% quando são feitas enquanto o semestre ainda está a decorrer. Além disso, a qualidade das previsões para as notas negativas, que são o real foco desta metodologia, atinge uma *accuracy* de 86% no final do semestre e uma *accuracy* média de 82% durante o decorrer do semestre. Os resultados obtidos no estudo são bastante promissores e provam que é possível criar uma sistema de alertas bastante preciso para ser usado por professores e estudantes, que pode ser integrado com qualquer disciplina, e que é dependente de uma plataforma de LMS.

6.1 Resposta às Questões de Investigação

Nesta secção, são dadas respostas às questões de investigação, referidas no capítulo 1.

Q1) Relativamente à primeira questão, **é possível prever o resultado final de cada estudante com base, nas interações entre este e a plataforma?**

A resposta é sim é possivel. Foi feito um agrupamento de classificações em três categorias: negativas, instáveis e positivas. Como foi demonstrado pelos resultados obtidos no capítulo 5 o modelo é bastante preciso a obter classificações, é de notar os valores elevados de *F-score*, *accuracy*, *precision* e *recall*, nomeadamente quando se trata de prever classificações do tipo A.

Q2) É possível saber-se com antecedência, antes de terminar o semestre, se um estudante vai reprovar à unidade curricular?

Sim, parcialmente, os resultados obtidos, no capítulo 5, demonstram que mesmo no início do semestre o modelo é fidedigno a realizar previsões. Sendo que a qualidade do modelo evolui com o aumento do número de dados que este tem disponível. Tudo isto indica que o modelo é capaz de avisar correctamente, com antecedência, caso o estudante se encontre em risco de reprovar à unidade curricular.

Q3) Que tipo de modelo preditivo poderá ser usado?

Com base no estudo de outras publicações, no capítulo 3, é sugerido por vários autores que as *Support Vector Machine*s (SVMs), são de facto as mais eficazes a realizar previsões. Contudo, este estudo usou uma árvore de decisão, e os resultados obtidos mostram que este algoritmo pode ser considerado uma boa ferramenta para realizar previsões.

Q4) Existe alguma forma de calcular quanto tempo é que o estudante passou numa atividade do Moodle?

No capítulo 4 é descrito detalhadamente a metodologia, para o cálculo da duração de sessão de uma atividade usando os *logs* do Moodle. Foi criado um esquema (4.3) e o pseudo-código do algoritmo criado (4.2.4), de forma a demonstrar como este processo foi feito.

Q5) Será possível criar uma metodologia capaz de comparar o percurso académico de dois estudantes e no final atribuir um valor a essa comparação?

Sim, no capítulo 4 é descrito todo o processo para comparar o percurso académico de dois estudantes. Para realizar as comparações foram necessárias quatro fases. Primeiro realizar a conversão, das atividades efetuadas num dia para uma única *string* usando *Run-Length Encoding* (RLE). De seguida, a conversão da *string* obtida, para um ponto num espaço com 10-dimensões. Depois é realizado o cálculo da distância Euclidiania, para identificar o estudante com o qual se deve fazer a comparação, assumindo que este obteve uma classificação superior a 16. Por fim são aplicadas duas fórmulas matemáticas criadas (fórmulas 4.1 e 4.2) para atribuir um valor numérico à comparação feita.

6.2 Contribuições para a Comunidade Cientifica

6.2.1 Resumo das Contribuições

Esta dissertação originou as seguintes contribuições:

1) Criação de uma metodologia, desenvolvida, para o cálculo da duração de sessão das observações registadas nos ficheiros de *log* do Moodle;

2) A criação de uma árvore de decisão capaz de prever qual a classificação final, de um estudante, através da análise do *dataset* em conjunto com as variáveis independentes;

3) As fórmulas aplicadas, nomeadamente as fórmulas 4.1 e 4.2, para o cálculo da similaridade, que permitem quantificar o fator de similaridade entre dois percursos académicos de estudantes diferentes, que pertencem à mesma disciplina;

4) A identificação das *features* que foram usadas como variáveis independentes e ajudaram no processo de previsão do algoritmo;

5) A metodologia que engloba todo o processo, desde os dados iniciais ao produto final, capaz de realizar previsões.

6.2.2 Publicações

É importante referir algumas das publicações científicas que surgiram dos estudos realizados por esta dissertação. Todos os artigos criados são listados a seguir.

1) *Bruno Cabral and Álvaro Figueira . Preventing failures by predicting students' grades throughan analysis of logged data of online interactions. InProceedings of the 11th International JointConference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*

-Volume 1: KDIR,, pages 491–499. INSTICC, SciTePress, 2019d. ISBN: 978-989-758-382-7.doi:10.5220/0008356604910499..
Apresentado em Setembro de 2019, na conferência KDIR19 -11th International Conference on Knowledge Discovery and Information Retrieval. Neste artigo é explicado, detalhadamente, toda a metodologia que conduz à criação de um modelo capaz de prever classificações no final do semestre (Cabral e Figueira, 2019a).

2) *Bruno Cabral and Álvaro Figueira. On the development of a model to prevent failures, built from interactions with moodle. 18th International Conference on Web-Based Learning, Springer Lecture Notes in Computer Science (LNCS), Set 2019.*
Este artigo foi aceite na conferência ICWL19 - 8th International Conference on Web-Based Learning que decorreu em Setembro de 2019. Foca-se na relação entre as *features* identificadas e as atividades educativas de um modelo capaz de prever classificações (Cabral e Figueira, 2019b).

3) *Bruno Cabral and Álvaro Figueira. A machine learning model to early detect low performing students from lms logged interactions. 3th edition of International Conference Europe Middle East & North Africa On Information System Technology and Learning Researchs, Springer Lecture Notes in Computer Science (LNCS), Nov 2019.*
Publicado na conferência EMENA-ISLT19 - 3th edition of International Conference Europe MiddleEast & North Africa On Information System Technology and Learning Researchs, a ser realizada em Novembro. Com este artigo são explicados os avanços feitos à metodologia, nomeadamente a capacidade de realizar previsões ao longo do semestre. O seu principal tema, são as previsões feitas com um número reduzido de dados e o impacto, que os resultados obtidos, podem ter na possível implementação deste modelo com um sistema capaz de emitir alertas ao longo do semestre (Cabral e Figueira, 2019c).

6.3 Fragilidades

A metodologia explicada encontra-se adaptada à disciplina DPI1001, apesar da estrutura do modelo ser facilmente adaptada para que este seja aplicado a outras disciplinas, que façam o uso extensivo do Moodle.

Contudo, uma fragilidade da metodologia descrita é não poder ser aplicada a disciplinas que não usem as ferramentas disponibilizadas pelo Moodle, como o Fórum, entrega de trabalhos e testes. Quando uma disciplina não faz o uso na integra de todas as capacidades oferecidas pelo Moodle, existem poucas observações no ficheiro de log, que não estejam relacionadas com o acesso a uma atividade ou *download* de um ficheiro. Como tal, o modelo preditivo passa a ter poucas *features* disponíveis para fazer as previsões. O facto de existirem poucas *features* para se usar no modelo preditivo, reduz a capacidade de previsão do modelo.

6.4 Trabalho Futuro

Para trabalho futuro é de considerar uma expansão que permita a integração do modelo num sistema de alertas. Também se deve garantir a capacidade de generalização formal das *features* a outras disciplinas. Por fim, realizar testes que consistem na aplicação do modelo ao longo do ano letivo de uma disciplina e avaliar os resultados obtidos pelo modelo nas diferentes etapas do ano letivo, de forma a determinar o seu desempenho numa situação considerada real.

Bibliografia

A. K. Jain , M. N. Murty , and P. J. Flynn . Data clustering: A review. *ACM Comput. Surv.*, 31 (3):264–323, sep 1999. ISSN: 0360-0300.

Adilson Vahldick , António José Mendes , and Maria José Marcelino . Learning analytics model in a casual serious game for computer programming learning. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 176 LNICST:36–44, Dec 2017. ISSN: 18678211. doi:10.1007/978-3-319-51055-2_6.

Alberto Fernández , Salvador García , Francisco Herrera , and Nitesh V. Chawla . Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *J. Artif. Int. Res.*, 61(1):863–905, January 2018. ISSN: 1076-9757.

Alex J. Smola and Bernhard Schölkopf . A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, Ago 2004. ISSN: 1573-1375. doi:10.1023/B:STCO.0000035301.49549.88.

Álvaro Figueira . Predicting results from interaction patterns during online group work. CP 10.1007/978-3-319-24258-3_33, CRACS & INESC TEC – University of Porto, Set 2015.

Álvaro Figueira . Mining moodle logs for grade prediction: A methodology walk-through. *Sociologia, Problemas e Praticas*, 81:115–140, Jun 2016. doi:10.1145/3144826.3145394.

Álvaro Figueira . Predicting grades by principal component analysis: A data mining approach to learning analyics. *Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality*, pages 44:1–44:8, Jun 2017.

Ana Paula Lopes . Teaching with moodle in higher eduction. In *INTED2011 Proceedings*, 5th International Technology, Education and Development Conference, pages 970–976. IATED, March 2011. ISBN: 978-84-614-7423-3.

Bruno Cabral and Álvaro Figueira . Preventing failures by predicting students' grades through an analysis of logged data of online interactions. In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 1: KDIR,*, pages 491–499. INSTICC, SciTePress, 2019a. ISBN: 978-989-758-382-7. doi:10.5220/0008356604910499.

Bruno Cabral and Álvaro Figueira . On the development of a model to prevent failures, built from interactions with moodle. *18th International Conference on Web-Based Learning*, Set 2019b.

Bruno Cabral and Álvaro Figueira . A machine learning model to early detect low performing students from lms logged interactions. *3th edition of International Conference Europe Middle East & North Africa On Information System Technology and Learning Researchs*, Nov 2019c.

C. R. Maurer , Rensheng Qi , and V. Raghavan . A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):265–270, Fev 2003. ISSN: 0162-8828. doi:10.1109/TPAMI.2003.1177156.

Carlos Martins , Carla Morais , and Luciano Moreira . Images of the moodle: Social representations of higher education teachers and students. In *EDULEARN19 Proceedings*, 07 2019. doi:10.21125/edulearn.2019.0756.

Carolina Costa , Helena Alvelos , and Leonor Teixeira . The use of moodle e-learning platform: A study in a portuguese university. *Procedia Technology*, 5:334 – 343, 2012. ISSN: 2212-0173. 4th Conference of ENTERprise Information Systems – aligning technology, organizations and people (CENTERIS 2012).

Charles X. Ling and Chenghui Li . Data mining for direct marketing: Problems and solutions. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 73–79. AAAI Press, 1998.

Christopher J.C. Burges . A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, Jun 1998. ISSN: 1573-756X. doi:10.1023/A:1009715923555.

Christopher M. Bishop . *Pattern Recognition and Machine Learning*. Springer, Mai 2006.

Colin Shearer . The crisp-dm model: The new blueprint for data mining. *Journal of Data Warehousing Volume 5 Number 4 Fall 2000*, 5:13–22, Abr 2000.

Corinna Cortes and Vladimir Vapnik . Support-vector networks. *Machine Learning*, 20(3): 273–297, Sep 1995. ISSN: 1573-0565. doi:10.1007/BF00994018.

Cristóbal Romero and Sebastian Ventura . Web usage mining forpredicting final marks ofstudents that use moodlecourses. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40:601 – 618, 12 2010. doi:10.1109/TSMCC.2010.2053532.

Cristóbal Romero , Pedro G. Espejo , Amelia Zafra , Jose Raul Romero , and Sebastian Ventura . Web usage mining for predicting final marks of students that use moodle courses. *Computer Applications in Engineering Education*, 21(1):135–146, 2013. doi:10.1002/cae.20456.

David M. Magerman . Statistical decision-tree models for parsing. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, ACL '95, pages 276–283. Association for Computational Linguistics, 1995.

David M. W. Powers . Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation,. *Journal of Machine Learning Technologies*, 2:37–63, Fev 2011.

DGEEC Direção-Geral de Estatísticas da Educação e Ciência . Perfil do docente 2016/17. Online, Set 2018.

Dong-Hui Xu , Arati S Kurani , Jacob D Furst , and Daniela S. Raicu . Run-length encoding for volumetric texture. *The 4th IASTED International Conference on Visualization, Imaging, and Image Processing*, 27:452–458, Mai 2004.

Dragan Gašević , Shane Dawson , Tim Rogers , and Danijela Gasevic . Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting academic success. *The Internet and Higher Education*, 28:68 – 84, 2016. ISSN: 1096-7516.

DRE Diário da República nº 116/2005 Série II . Despacho n.º 13584/2005 (2.ª série). Online, Jun 2005. 6-Jan-2019.

Eliana Santana Lisbôa , Gláucia Helena Sales Teixeira , Anabela Gomes de Jesus , António Manuel Leitão Macedo Varela , and Clara Pereira Coutinho . Computador e a internet como instrumentos pedagógicos : estudo exploratório com professores de duas escolas do norte de portugal. *Universidade do Minho*, Sep 2009.

Erhard Rahm , , and Hong Hai Do . Data engineering - special issue on data cleaning. *Data Engineering*, 23:3–13, Dec 2000.

Fabian Pedregosa , Gaël Varoquaux , Alexandre Gramfort , Vincent Michel , Bertrand Thirion , Olivier Grisel , Mathieu Blondel , Peter Prettenhofer , Ron Weiss , Vincent Dubourg , Jake VanderPlas , Alexandre Passos , David Cournapeau , Matthieu Brucher , Matthieu Perrot , and Edouard Duchesnay . Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, Oct 2011.

H. Breu , J. Gil , D . Kirkpatrick , and M. Werman . Linear time euclidean distance transform algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):529–533, Mai 1995. ISSN: 0162-8828. doi:10.1109/34.391389.

Hao Huang , Haihua Xu , Xianhui Wang , and Wushour Silamu . Maximum f1-score discriminative training criterion for automatic mispronunciation detection. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 23:787–797, Abr 2015. doi:10.1109/TASLP.2015.2409733.

Igor Felix , Ana Paula Ambrosio , Jacques Duilio , and Eduardo Simões . Predicting student outcome in moodle. *CASHE: Conference of Academic Success in Higher Education*, pages 14–15, Fev 2019.

Ihab Ahmed Najm , Alaa Khalaf Hamoud , Jaime Lloret , and Ignacio Bosch . Machine learning prediction approach to enhance congestion control in 5g iot environment. *Electronics*, 8(6), 2019. ISSN: 2079-9292. doi:10.3390/electronics8060607.

Ivana Đurđević Babić . Machine learning methods in predicting the student academic motivation. *Croatian Operational Research Review*, 8:443–461, Abr 2017. ISSN: 18480225. doi:10.17535/crorr.2017.0028.

Joaquim Duarte and Maria Gomes . Práticas com a moodle em portugal. *VII Conferência Internacional de TIC na Educação*, 09 2011.

João Fernandes . Moodle nas escolas portuguesas - números, oportunidades, ideias. *Caldas Moodle 2008. Educom - Associação Portuguesa de Telemática Educativa*, September 2008.

Jun Wang and Ying Tan . Efficient euclidean distance transform algorithm of binary images in arbitrary dimensions. *Pattern Recognition*, 46(1):230–242, 2013. ISSN: 0031-3203.

M. Delgado Calvo-Flores , E. Gibaja Galindo , M. C. Pegalajar Jímenez , and O. Pérez Piñeiro . Predicting students' marks from moodle logs using neural network models. *Current Developments in Technology-Assisted Education*, pages 586–590, 2006.

M.A. Friedl and C.E. Brodley . Decision tree classification of land cover from remotely sensed data. *Remote Sensing of Environment*, 61(3):399 – 409, 1997. ISSN: 0034-4257.

Max Khum and Kjell Johnson . *Applied Predictive Modeling*. Springer, Mai 2016. ISBN: 978-1-4614-6848-6. doi:10.1007/978-1-4614-6849-3.

Mehmed Kantardzic . *Data Mining Concepts, Models, Methods, and Algorithms*. IEEE Press, Ago 2011. ISBN: 9780470890455.

Mi I. López , Jm M. Luna , C. Romero , and S. Ventura . Classification via clustering for predicting final marks based on student participation in forums. *Proceedings of the 5th International Conference on Educational Data Mining*, pages 4–7, 2012.

Monika Simjanoska , Marjan Gusev , and Ana Madevska Bogdanova . Intelligent modelling for predicting students' final grades. *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2014 - Proceedings*, pages 1216–1221, Mai 2014. doi:10.1109/MIPRO.2014.6859753.

Naveen Venkat , Sahaj Srivastava , and Lakshya Garg . Predicting student grades using machine learning. Technical report, BITS Pilani, 11 2018. doi:10.13140/RG.2.2.21516.77449.

Nuno Gil Fonseca , António José Mendes , and Luís Macedo . *CodeInsights - Monitorização do desempenho de alunos de programação*. Bartleby, Nov 2016.

Paulo Pimentel . Impacto da plataforma moodle nas escolas de famalicão : um estudo de caso. *Universidade do Minho*, pages 1–132, Jun 2009.

Paulo Coelho Dias , Nuno De Almeida Alves , Pedro Abrantes , and Carla F. Rodrigues . Utilização da plataforma moodle em portugal: Moodle nas escolas do ensino básico e secundário em portugal. *Sociologia, Problemas e Praticas*, 81:115–140, Jun 2016. ISSN: 08736529. doi:10.7458/SPP2016813145.

Platform Moodle . History - moodledocs. Online, Mai 2019. Acedido 27 de Maio de 2019.

Rianne Conijn , Chris Snijders , Ad Kleingeld , and Uwe Matzat . Predicting student performance from lms data: A comparison of 17 blended courses using moodle lms. *IEEE Transactions on Learning Technologies*, 10(1):17–29, Jan 2017. ISSN: 1939-1382. doi:10.1109/TLT.2016.2616312.

Robert Cooley , Bamshad Mobasher , and Jaideep Srivastava . Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, Feb 1999. doi:10.1007/BF03325089.

Rüdiger Wirth . Crisp-dm : Towards a standard process model for data mining. *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, pages 29–39, 2000. ISSN: 1092-6208.

S. C. Hinds , J. L. Fisher , and D. P. D'Amato . A document skew detection method using run-length encoding and the hough transform. In *[1990] Proceedings. 10th International Conference on Pattern Recognition*, volume 1, pages 464–468 vol.1, Jun 1990. doi:10.1109/ICPR.1990.118147.

S. R. Safavian and D. Landgrebe . A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):660–674, May 1991. ISSN: 0018-9472. doi:10.1109/21.97458.

Santiago Iglesias-Pradas , Carmen Ruiz-de-Azcárate , and Ángel F. Agudo-Peregrina . Assessing the suitability of student interactions from moodle data logs as predictors of cross-curricular competencies. *Computers in Human Behavior*, 47:81 – 89, 2015. ISSN: 0747-5632.

Shichao Zhang , Chengqi Zhang , and Qiang Yang . Data preparation for data mining. *Applied Artificial Intelligence*, 17(5-6):375–381, 2003. doi:10.1080/713827180.

Soohyun Nam Liao , Daniel Zingaro , Kevin Thai , Christine Alvarado , William G. Griswold , and Leo Porter . A robust machine learning technique to predict low-performing students. *ACM Trans. Comput. Educ.*, 19(3):18:1–18:19, January 2019. ISSN: 1946-6226. doi:10.1145/3277569.

Srecko Joksimovic , Dragan Gasevic , Thomas M. Loughin , Vitomir Kovanovic , and Marek Hatala . Learning at distance: Effects of interaction traces on academic achievement. *Computers & Education*, 87:204–217, 07 2015. doi:10.1016/j.compedu.2015.07.002.

Thorsten Joachims . Text categorization with support vector machines: Learning with many relevant features. In Nédellec Claire and Rouveirol Céline , editors, *Machine Learning: ECML-98*, pages 137–142, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. ISBN: 78-3-540-69781-7.

Tomas Escobar-Rodriguez and Pedro Monge-Lozano . The acceptance of moodle technology by business administration students. *Computers & Education*, 58(4):1085 – 1093, 2012. ISSN: 0360-1315.

Vladimir Vapnik . *The Nature Of Statistical Learning Theory*, volume 6. Springer, 01 1995. doi:10.1007/978-1-4757-2440-0.